Titre: Title:	Ingénierie de trafic et réservation de ressources dans les réseaux cellulaires IPv6
Auteur: Author:	Stéphane Ouellette
Date:	2006
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Ouellette, S. (2006). Ingénierie de trafic et réservation de ressources dans les réseaux cellulaires IPv6 [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/8930/

UNIVERSITÉ DE MONTRÉAL

INGÉNIERIE DE TRAFIC ET RÉSERVATION DE RESSOURCES DANS LES RÉSEAUX CELLULAIRES IPv6

STÉPHANE OUELLETTE
DÉPARTEMENT DE GÉNIE INFORMATIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION DU DIPLÔME DE
MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE INFORMATIQUE)
AVRIL 2006

Your file Votre référence
ISBN: 978-0-494-17963-5
Our file Notre référence
ISBN: 978-0-494-17963-5

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

INGÉNIERIE DE TRAFIC ET RÉSERVATION DE RESSOURCES DANS LES RÉSEAUX CELLULAIRES IPv6

présenté par : Stéphane Ouellette
en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées
a été dûment accepté par le jury constitué de :

M. QUINTERO Alejandro, Doct., président.
M. PIERRE Samuel, Ph.D., membre et directeur de recherche.
M. BENSLIMANE Abderrahim, Doct., membre.

À mon épouse Isabelle,
pour sa compréhension et son soutien.

Remerciements

J'aimerais tout d'abord remercier mon directeur de recherche, Samuel Pierre, pour m'avoir soutenu, conseillé et guidé tout au long de mon projet de recherche. Il m'a inculqué la rigueur scientifique dont je fais maintenant preuve.

Je voudrais aussi souligner la contribution très importante de Ericsson Research Canada, en particulier Laurent Marchand et Yves Lemieux, pour leurs précieux conseils et leur soutien tout au long de mes travaux de recherche.

Enfin, j'aimerais remercier mes collègues et amis du *LAboratoire de recherche en Réseautique et Informatique Mobile* (LARIM) pour leur collaboration. J'ai passé en leur compagnie de nombreuses heures à discuter des travaux de recherche de tous et chacun. Ils ont ainsi suscité en moi l'intérêt pour la recherche, le tout dans une ambiance des plus chaleureuses.

Résumé

Depuis quelques années, des applications de téléphonie IP ont émergé. Des noms tels que *Skype* et *Netmeeting* sont aujourd'hui connus du grand public. Toutefois, l'Internet actuel est mal adapté à de telles applications puisqu'il ne garantit aucune *Qualité de Service* (QdS) aux usagers. La gestion de QdS met à profit les mécanismes qui garantissent les paramètres d'une connexion, peu importe la charge du réseau.

L'absence de QdS a pour conséquence une dégradation de la communication qui est inévitable lorsque le réseau est fortement congestionné. La congestion est un phénomène propre aux réseaux dits à commutation de paquets conventionnels en raison du partage des ressources entre les entités. À l'opposé, les systèmes téléphoniques traditionnels, dits à commutation de circuits, ne souffrent pas de tels problèmes puisque les ressources sont réservées pour toute la durée de la communication.

Les premiers systèmes de téléphonie IP ayant une QdS comparable au réseau téléphonique commuté, sont apparus dans les bureaux de grandes compagnies supportant déjà une infrastructure réseau étendue. Une portion de la bande passante totale est alors réservée aux communications téléphoniques. Par contre, ces systèmes sont fixes et souvent limités à un seul édifice.

Le déploiement du protocole IPv6 devrait mener à une nouvelle ère pour l'Internet. En effet, ses nombreux avantages comparativement à IPv4, dont le support d'un identificateur de flot pour les communications temps-réel, l'immense espace d'adressage de 128 bits et les mécanismes d'autoconfiguration d'adresse rendent IPv6 beaucoup plus attrayant pour les communications temps-réel. De plus, le support de la mobilité permet aux usagers de se déplacer d'un réseau d'accès à un autre, tout en demeurant accessibles à partir d'une adresse permanente, localisée dans leur réseau résidentiel.

La mobilité des usagers occupera une partie de plus en plus importante de l'ensemble des communications au cours du prochain siècle. Déjà aujourd'hui, on voit apparaître des applications de tous genres pour les appareils cellulaires tirant profit de la géolocalisation et du profil de l'usager afin d'offrir des services personnalisés.

Un autre aspect de la mobilité des usagers à promouvoir est le concept d'itinérance globale. Ce concept vise à rendre un usager capable de profiter de tous les services auxquels il a souscrit dans son réseau résidentiel, et ce n'importe où dans le monde.

Un grand défi de la mobilité est d'offrir la capacité de changer de point d'accès tout en préservant la QdS des connexions en cours. C'est pourquoi nous traiterons plus particulièrement des relèves intra-domaines dans les réseaux d'accès cellulaires.

La rédaction de ce mémoire a été motivée par deux objectifs. Le premier est d'aboutir à une proposition solide de protocole de réservation de ressources qui soit adapté à un environnement mobile. Ce protocole permettra d'appliquer les concepts propres à l'ingénierie de trafic à l'intérieur du réseau d'accès. Le second objectif est de tirer profit du champ `Flow Label` de IPv6 afin de construire des chemins commutés par étiquette.

Une tendance actuelle de l'industrie est de développer des équipements dans lesquels la classification des flots de données serait effectuée uniquement au routeur à l'entrée du réseau. En conséquence, les paquets seraient étiquettés selon la classe de service qui leur est assignée.

Une classe de service regroupe un ensemble de paramètres de QdS qui doivent être appliqués à un flot de données. Dans un réseau MPLS, la classe de service est représentée par une seule valeur numérique. Cette valeur est utilisée par le routeur suivant, sur le chemin du flot de données, pour identifier le port de sortie ainsi que les politiques de QdS à appliquer.

Dans le but de permettre l'évolutivité du réseau, les valeurs des étiquettes n'ont qu'une signification locale sur un lien. En conséquence, la même valeur peut être utilisée sur un autre lien, sans risque de confusion. De plus, chaque routeur remplace la valeur de l'étiquette par une autre valeur qui a une signification uniquement pour le prochain routeur qui se trouve sur le chemin emprunté par le flot de données.

Le réseau Internet est basé sur la commutation de paquets. Le principal avantage de cette approche est l'optimisation de l'utilisation du canal sur lequel toutes les connexions transitent. Par contre, cette approche a comme inconvénient d'être mal adaptée au trafic temps-réel et puisque la variabilité du délai de bout en bout est importante, les paquets peuvent atteindre leur destination dans le désordre et peuvent même être perdus en raison de la congestion du réseau.

Dans un autre ordre d'idée, les réseaux à commutation de circuits, tel que le réseau téléphonique commuté, offrent l'avantage de réserver des ressources lors de l'établissement de chaque connexion. De plus, ces ressources sont garanties pour la durée complète de la connexion en question et les données sont livrées dans l'ordre où elles ont été émises. En contrepartie, un haut niveau de congestion du réseau se traduit

par l'impossibilité d'établir une nouvelle connexion. Enfin, les réseaux à commutation de circuits s'adaptent mal aux pannes de réseau qui mènent généralement à la perte de la connexion.

Il est clair qu'une approche hybride permettrait de tirer profit des modèles de commutation de paquets et de circuits. En conséquence, l'établissement de circuits virtuels permettrait de garantir les réservations de ressources tout en évitant le gaspillage de ces dernières lorsqu'elles sont faiblement utilisées.

L'approche des circuits virtuels fut retenue dans le cadre de notre proposition de solution qui est consituée de deux volets. Le premier volet consiste en une proposition d'architecture de transport tirant profit du champ `Flow Label` de IPv6 afin de construire des chemins commutés par étiquette. Le second volet de la proposition de solution est l'élaboration d'un nouveau protocole combinant à la fois les réservations de ressources ainsi que la distribution d'étiquettes. Ce protocole est fortement inspiré des protocoles RSVP-TE et HPMRSVP. En conséquence, le nom qui lui a été attribué est HPMRSVP-TE.

Suite à l'élaboration du protocole HPMRSVP-TE, nous avons construit un modèle analytique afin d'évaluer les probabilités de blocage, d'interruption forcée d'une session en cours et de compléter une session avec succès.

Ensuite, nous avons procédé à une étude de coûts en ressources et en temps, afin de dresser un profil des performances de HPMRSVP-TE. Ce modèle donne une bonne idée du temps requis pour effectuer les tâches courantes telles que les réservations initiales de ressources et le déplacement de ces réservations lors d'une relève intra-domaine.

La contribution majeure de ce travail de recherche réside dans la combinaison des avantages offerts par les protocoles HPMRSVP, quant aux déplacements des réservations de ressources lors des relèves de niveau 3, et RSVP-TE, pour la distribution d'étiquettes et la réservation des ressources. Il en résulte un protocole capable d'appliquer les concepts d'ingénierie de trafic au réseau d'accès qu'il supporte.

Une seconde contribution significative vise à corriger une lacune importante dans l'architecture IPngLS qui a été retenue pour remplacer MPLS à titre d'architecture de transport. En effet, le défaut majeur de cette architecture est de détruire la valeur du champ `Flow Label` de IPv6 pour chaque paquet traité. Nous avons remédié à ce problème grâce à une modification originale d'un objet RSVP qui identifie une session temps-réel de façon unique. En conséquence, la valeur originelle du champ

`Flow Label` est récupérée à la sortie du réseau, assurant ainsi la transparence de ce dernier.

Au terme de ce projet, nous avons identifié quelques avenues possibles pouvant mener à des améliorations du protocole HPMRSVP-TE. Un premier ensemble d'améliorations qui a été identifié vise à simplifier le traitement des requêtes du protocole au niveau des routeurs. En effet, la première amélioration consiste à créer un message capable de regrouper le contenu de plusieurs messages. Le but de cette démarche est de réduire la charge de travail des routeurs. Une autre avenue consisterait à ajouter un identificateur unique qui permettrait aux routeurs de reconnaître les messages de rafraîchissement sans devoir examiner ceux-ci en détail.

Un second ensemble d'améliorations provient d'une extension récente au protocole RSVP-TE qui permet d'allouer des ressources supplémentaires afin de réserver des chemins de contournement, en prévision d'une panne.

Abstract

During the last few years, a number of phone applications based on IP have emerged. The general public is now aware of the existence of application names such as *Skype* and *Netmeeting*. Unfortunately, today's Internet doesn't deal appropriately with these applications because it can't guarantee any *Quality of Service* (QoS) to its subscribers. QoS management takes advantage of mechanisms that guarantee a connection's parameters regardless of the actual network load.

The lack of QoS management causes a deterioration of the communication when the network is heavily congested. Congestion is a phenomenon encountered in traditional packet-switched networks as a consequence of sharing the available resources between the network nodes. On the other hand, circuit-switched networks such as the traditional telephone system don't suffer from these problems because the needed resources are allocated for the duration of the call.

The first telephone systems based on IP that had a QoS similar to their circuit-switched counterparts, appeared in large companies' offices that already had an extensive computer network coverage. A fraction of the total bandwidth was dedicated to phone communications. On the other hand, these systems were fixed and usually limited to a single building.

The deployment of the IPv6 protocol shall lead the Internet to a new era. In fact, IPv6 has a number of features that were absent in IPv4, such as a flow label identifier for real-time sessions, a huge 128-bit address space and an autoconfiguration mode that make IPv6 more suitable for real-time communications. Additionally, mobility support allows subscribers to roam from one access network to another, while remaining accessible from a permanent address, located in their home network.

Mobile communications will increase their share of the total number of communications during the next century. Today, cellular phone applications taking advantage of geopositioning and of the subscriber's profile are growing in popularity, making it possible for the service providers to offer personalized services.

Another aspect of user mobility that should be extended is the concept of global roaming. Global roaming allows users to take advantage of all of the services they subscribed to in their home network, anywhere in the world.

A great challenge of user mobility is to provide the ability to roam from one access point to another while preserving the QoS of real-time sessions. As such, we will deal more specifically with intra-domain handovers in cellular access networks.

The reasons that motivated the redaction of this document aim two main goals. The first one was to crystalize a comprehensive proposal for a resource reservation protocol that is well-suited for a mobile environment. This protocol will allow the application of traffic engineering concepts inside the access network. The second one is to take advantage of the IPv6 Flow Label field to build label-switched paths.

At this time, there is trend within the industry to design equipments in which flow classification would be done only by the edge router. As a consequence, packets would be labeled according to their assigned service class.

A service class groups a set of QoS parameters that must be applied to a data flow. In a MPLS network, the service class is represented by a single numeric value. This value is then used by the following router, on the data flow's path, to identify the output port and QoS policies to apply.

In order to promote the network's scalability, label values only have a local significance on a given link. As a consequence, the same label value can be reused on a different link without any risk of confusion. Also, each router replaces an incoming label value with another that has a local significance only for the following router on the path of the data flow.

The Internet is a packet-switched network. The main advantage of this approach is the optimal use of the channel on which all connections transit. But, this approach is unsuitable for real-time traffic because of its important end-to-end jitter, packets can be delivered out of order or can even be lost because of network congestion.

On the other hand, circuit-switched networks such as the PSTN reserve resources when a connection is established. Also, these resources are reserved for the duration of the session and the carried data is delivered in the same order it was sent. But there is a drawback: a high level of congestion makes it impossible to establish new connections. At last, circuit-switched networks cannot easily adapt to network failures and would usually drop the connection.

It is clear that a hybrid approach would benefit from the packet-switched and circuit-switched network models. Consequently, the establishment of virtual circuits would guarantee the resource reservations while avoiding to waste them when they are lightly used.

The virtual circuits approach was chosen for our final proposal which is made of two aspects. The first aspect was to propose a transport architecture which benefits from the use of the IPv6 `Flow Label` field to build label-switched paths. The second aspect consists in the design of a new resource reservation and label distribution protocol. This protocol is highly inspired from the RSVP-TE and HPMRSVP protocols, therefore it was named HPMRSVP-TE.

Following the design of the HPMRSVP-TE protocol, we built an analytical model in order to evaluate the blocking, forced interruption and completion probabilities.

Then, we evaluated the costs in resources and time in order to determine the performance profile of HPMRSVP-TE. This model provides us with an estimate of the time required to perform routine tasks such as the initial resource reservation and the move of these reservations during an intra-domain handover.

The major contribution of the research project resides in the combination of HPMRSVP's benefits, regarding the mobility of resource reservations, and RSVP-TE, for label distribution and resource reservation. The result is a protocol that can apply traffic engineering concepts to the access network it supports.

Another important contribution of this work solves a serious problem found in the IPngLS architecture that was chosen to replace MPLS as the transport architecture. In fact, the major drawback of this architecture is that the IPv6 `Flow Label` field's value is lost for each packet processed. We solved this problem by using a creative modification of a RSVP object that uniquely defines a real-time session. As a consequence, the original value of the `Flow Label` field is restored at the egress node, assuring the transparency of the network.

At the term of this project, we identified a few ways to improve the HPMRSVP-TE protocol. The first set of modifications that were identified include mechanisms to simplify the processing of requests by the routers. In fact, the first improvement is to create a message capable of grouping many independant messages into a single bundle, thus reducing the work load for the routers. Another improvement would be to add a unique message identifier so that refresh messages could be easily recognized by routers without having to inspect them thoroughly.

A second set of improvements originates from a recent extension to the RSVP-TE protocol that allows additional resources to be reserved in order to build failover paths.

Table des matières

Liste des tableaux

Liste des figures

Liste des sigles et abréviations

6LSA	IPv6 Label Switching Architecture
6LSP	6LSA Label Switched Path
6LSR	6LSA Label Switching Router
AAA	Authentication, Authorization and Accounting
AES	Access Edge Site
AR	Access Router
AS	Autonomous System
ATM	Asynchronous Transfer Mode
BES	Border Edge Site
CN	Correspondent Node (RFC3344)
CoA	Care-of Address (RFC3344)
DiffServ	Differentiated Services (RFC2475)
DS	Differentiated Services field (RFC2474)
DSCP	Differentiated Services Code Point (RFC2474)
FBACK	Fast Binding Acknowledgement (RFC4068)
FBU	Fast Binding Update (RFC4068)
FEC	Forwarding Equivalence Class
F-HMIPv6	Fast-handover for Hierarchical Mobile IPv6
FMIPv6	Fast-handover for Mobile IPv6 (RFC4068)
FNA	Fast Neighbor Advertisement (RFC4068)
HA	Home agent (Mobile IP, RFC3344)
HACK	Handover Acknowledge (RFC4068)
HI	Handover Initiate (RFC4068)
HoA	Home address (RFC3344)
HMIPv6	Hierarchical Mobile IPv6 (RFC4140)
IANA	Internet Assigned Number Authority
ICMPv6	Internet Control Message Protocol for IPv6 (RFC2463)
IETF	Internet Engineering Task Force
IntServ	Integrated Services (RFC1633)
IP ou IPv4	Internet Protocol, version 4 (RFC791)

IPv6	Internet Protocol, version 6 (RFC2460)
LBACK	Local Binding Acknowledgement (RFC4140)
LBU	Local Binding Update (RFC4140)
LCoA	Local Care-of Address (RFC4140)
LSP	Label-Switched Path (RFC1633)
LSR	Label Switching Router (RFC1633)
MAP	Mobility Anchor Point (RFC4140)
MIPv6	Mobile IPv6 (RFC3775)
MN	Mobile Node (RFC3344)
MPLS	MultiProtocol Label Switching (RFC3031)
MSPEC	Mobile Specification (MRSVP)
MTU	Maximum Transfer Unit
NAR	New Access Router
OSI	Open Systems Interconnection
PAR	Previous Access Router
PrRtAdv	Proxy Router Advertisement (RFC4068)
PHB	Per-Hop Behavior (RFC2475)
PHOP	Previous Hop
QdS	Qualité de Service
RA	Router Advertisement (RFC3775)
RCoA	Regional Care-of Address (RFC4140)
RFC	Request For Comment
RS	Router Sollicitation (RFC3775)
RSVP	Resource ReSerVation Protocol (RFC2205)
RSVP-TE	RSVP for Traffic Engineering (RFC3209)
RTCP	Real-Time Control Protocol (RFC3550)
RTP	Real-Time Protocol (RFC3550)
RtSolPr	Router Solicitation for Proxy Advertisement (RFC4068)
SDP	Session Description Protocol (RFC2327)
SIP	Session Initiation Protocol (RFC3261)
TCP	Transmission Control Protocol (RFC793)
TOS	Type Of Service byte (RFC791, RFC1349)
TTL	Time-To-Live
UDP	User Datagram Protocol (RFC768)

VCI	Virtual Circuit Identifier (ATM)
VPI	Virtual Path Identifier (ATM)
VPN	Virtual Private Network
WAN	Wide Area Networks

CHAPITRE 1

INTRODUCTION

Le réseau Internet actuel voit apparaître de nouvelles applications dont les besoins sont très différents de ceux des premières années de son existence. Ces types d'applications requièrent une Qualité de Service (QdS) qui répond à des critères de débit, de délai maximal de bout-en-bout et de variation de délai (*jitter*). Par ailleurs, les fournisseurs de services tentent de maximiser l'utilisation de leurs infrastructures physiques par des mécanismes de classification et d'ingénierie de trafic.

Ce mémoire se penchera donc sur la création de chemins commutés par étiquette dans IPv6. Ainsi, il deviendra possible d'appliquer des concepts d'ingénierie de trafic. Nous concentrerons nos efforts dans les réseaux d'accès pour usagers mobiles.

Dans ce chapitre d'introduction, nous présenterons d'abord les définitions et concepts de base exposés dans ce mémoire. Ensuite, nous aborderons les éléments de la problématique qui justifient cette recherche et qui sont intimement liés aux objectifs de recherche décrits à leur suite. Enfin, nous exposerons le plan du mémoire, offrant ainsi un aperçu des chapitres à venir.

1.1 Définitions et concepts de base

Nous aborderons dans cette section les concepts de base requis à la compréhension des éléments de ce mémoire. Tout d'abord, nous commencerons par expliquer comment s'effectue le routage conventionnel des paquets dans un réseau IP.

Ensuite, nous discuterons des types d'applications que les usagers exécutent sur leur appareil et en quoi celles-ci requièrent certaines garanties qui imposent des contraintes différentes sur le réseau.

Dans un autre ordre d'idée, nous passerons en revue les aspects axés sur le réseau lui-même. En effet, nous expliquerons les concepts de QdS, de classification des flots de données et de chemins commutés par étiquette (*Label-Switched Path*, LSP). Nous verrons en quoi ces concepts ouvrent la voie à l'ingénierie de trafic.

1.1.1 Le routage conventionnel

Dans un réseau à commutation de paquets conventionnel, l'acheminement des paquets se base sur l'adresse destination inscrite dans l'en-tête du paquet.

En effet, chaque nœud du réseau possède une *table de routage* dont les entrées associent un préfixe de réseau et une métrique de coût à un port de sortie du nœud.

Pour un paquet donné, la sélection d'une entrée dans la table de routage est basée sur le plus long préfixe de réseau trouvé dans cette table. Dans le cas où deux préfixes de même longueur sont trouvés, on base la décision finale sur le coût minimal. La métrique de coût minimal indique en général le nombre de sauts pour atteindre le réseau indiqué dans l'entrée de la table de routage.

1.1.2 Les types d'applications

Les premières applications utilisées dans les réseaux IP n'avaient que de faibles contraintes quant au débit requis ou au délai de bout-en-bout : le trafic de données était majoritairement du courrier électronique et des transferts de fichiers. En conséquence, la priorité était accordée à l'acheminement des paquets avec succès.

Les futures applications comporteront de fortes exigences de délai et de débit qui devront être garanties par le réseau ; ces applications sont dites *temps-réel.*

La téléphonie IP et les vidéo-conférences sont des exemples d'applications *temps-réel.* En effet, le bon fonctionnement de ces applications dépend de la capacité du réseau à acheminer les données selon un débit garanti, tout en respectant un délai de bout-en-bout maximal et borné ainsi qu'une variation maximale et bornée du délai.

1.1.3 Le concept de Qualité de Service

La gestion de la QdS dans un réseau repose sur un ensemble de mécanismes par lesquels une application signifie au réseau ses besoins en ressources pour toute la durée de la connexion. Les besoins des flots de données peuvent être catégorisés selon quatre paramètres importants (voir Tanenbaum, 2002, sect. 5.4) :

- la fiabilité (acheminement des données avec succès) ;
- le délai de bout-en-bout de la source vers la destination ;
- la variation du délai de bout-en-bout (*jitter*) ;
- la bande passante requise (le débit des informations).

Étapes du processus de gestion de la QdS par les routeurs

Pour qu'un flot bénéficie d'une QdS différente de la QdS de base, la première étape est d'associer le flot à une classe de service selon ses caractéristiques. La seconde étape consiste à marquer les paquets du flot afin de faciliter leur identification par les routeurs concernés par le transfert de données. Enfin, la dernière étape consiste en l'acheminement de ces flots tout en respectant les critères de QdS.

La classification des flots de données consiste à associer chaque flot à l'une des classes de services disponibles. Ces dernières distinguent les flots selon leurs importances relatives et les garanties de service qui leur sont associées. La classification des flots de données s'effectue généralement à la source ou au routeur d'entrée (*ingress*).

Le marquage des paquets est généralement effectué par la source ou le routeur *ingress* du réseau. Dans un réseau IPv6, les champs `DSCP` et `Flow Label` sont utilisés à cette fin. Dans le cas où le marquage des paquets n'est pas effectué par ces champs, un *filtre de spécification de trafic* est transmis à tous les routeurs concernés par le transfert de données, par la voie d'un protocole de réservation de ressources tel que RSVP (voir RFC2205 de Braden *et al.*, 1997).

L'acheminement des flots tire profit d'un ensemble de mécanismes qui permettent d'atteindre une bonne QdS dans un réseau :

- le surdimensionnement du réseau pour accroître la capacité ;
- la régularisation des flots par une mémoire tampon à la réception (*buffering*) ;
- la régularisation des flots par la source (*traffic shaping*) ;
- le module de contrôle d'admission de chaque routeur vérifie que les requêtes de réservations de ressources peuvent être satisfaites.

1.1.4 Les chemins commutés par étiquette

Certaines architectures de réseaux de transport, telles que ATM et MPLS, effectuent la classification des flots de données au nœud *ingress* du réseau et associent à chacun d'eux une classe de service qui est encodée dans une *étiquette.*

Une *étiquette* est une valeur numérique qui identifie, pour un nœud donné, à la fois le port de sortie ainsi que le niveau de QdS dont le paquet doit bénéficier. L'étiquette

est habituellement contenue dans un en-tête qui précède l'en-tête de la couche réseau.

En conséquence, dans le cas des paquets IPv6, l'acheminement ne se base plus sur l'adresse destination du paquet et la QdS ne dépend plus des champs `DSCP` et `Flow Label`. La commutation d'étiquettes ne se base que sur la valeur de l'étiquette, réduisant le temps de traitement au niveau des routeurs et simplifiant ainsi leur design.

1.1.5 L'ingénierie de trafic

L'acheminement des paquets selon l'adresse destination seulement peut amener certains nœuds du réseau à être congestionnés alors que d'autres demeurent sous-utilisés. Une utilisation judicieuse des LSP offre la possibilité de choisir le chemin par lequel un flot de données transite vers sa destination.

L'ingénierie de trafic est donc la discipline qui vise à maximiser l'utilisation des ressources du réseau tout en respectant les requêtes de QdS formulées par les usagers. Elle a recours aux statistiques générées par les routeurs pour élaborer une polititique de balancement de la charge du réseau. La politique est appliquée par un humain ou un automate habilité à le faire.

1.1.6 L'architecture Mobile IPv6 et ses dérivées

L'architecture MIPv6 (Johnson *et al.*, 2004) permet à un nœud mobile (*Mobile Node*, MN) ayant une adresse IPv6 permanente dans son réseau résidentiel (*home network*), de se déplacer de façon transparente vers d'autres points d'accès. En effet, un nœud interlocuteur (*Correspondent Node*, CN) ne peut déterminer si le MN est situé dans son réseau résidentiel ou dans un réseau visité.

Le transparence du mécanisme de mobilité est assurée par un agent local (*Home Agent*, HA) qui intercepte les paquets destinés au MN et vérifie si celui-ci se trouve dans le réseau résidentiel ou en visite dans un autre réseau. Dans le premier cas, le paquet lui est directement acheminé. En contrepartie, lorsque le MN se trouve dans un autre réseau, le HA encapsule le paquet destiné au MN et l'achemine vers l'adresse temporaire associée à ce dernier (*Care-of Address*, CoA).

Le protocole MIPv6 supporte une optimisation de route qui évite aux paquets de transiter par le HA pour se rendre au MN ; le MN crée directement une association avec son CN et devient responsable de l'informer en cas de changement de CoA.

Un déplacement du MN d'un point d'accès à un autre porte le nom de *relève*. La relève peut être initiée par le MN ou le réseau et s'accompagne du renouvellement des associations que le MN entretient avec son HA et ses CN.

Le MN peut décider d'enclencher la relève suite à la réception de l'annonce d'un nouveau routeur d'accès spécifié par un message ICMPv6 (Conta et Deering, 1998; Narten *et al.*, 1998). La relève initiée par le réseau débute par l'envoi d'une annonce de routeur d'accès qui force le MN à changer de point d'accès.

Considérant les impacts négatifs de la relève sur la QdS, l'extension FMIPv6 (Koodli, 2005) a été développée afin de réduire la perte de paquets. La méthode employée consiste à rediriger les paquets en transit, pendant toute la durée de la relève, de l'ancien routeur d'accès (*Previous Access Router*, PAR) vers le nouveau routeur d'accès (*New Access Router*, NAR).

Hierarchical Mobile IPv6 (HMIPv6)

L'architecture HMIPv6 (Soliman *et al.*, 2005) est une amélioration de MIPv6 parce qu'elle réduit la signalisation en introduisant une hiérarchie de domaines de mobilité. En effet, introduit le concept de *Mobility Anchor Point* (MAP) qui a pour objectif de traiter différemment les deux cas de mobilité :

- la *micro-mobilité* qui suppose une relève intra-fournisseur de services ;
- la *macro-mobilité* qui implique une relève inter-fournisseurs de services.

Dans le cas où la relève ne s'effectue qu'entre deux points d'accès qui sont sous le contrôle du même MAP, le MAP n'informe pas le HA du déplacement du MN, réduisant ainsi la signalisation. Par contre, une relève inter-fournisseurs se produit de la même façon qu'en MIPv6.

Fast Handovers for Mobile IPv6 (FMIPv6)

Koodli (2005) a proposé un protocole afin de diminuer le risque de perte de paquets lors de la relève. L'approche préconisée est l'établissement d'un tunnel entre le PAR et le NAR afin d'y rediriger le trafic destiné au MN. Le NAR accumule les paquets et attend que le MN se reconnecte. Dès que le MN est reconnecté, ce dernier envoie un message FNA[1] au NAR pour lui annoncer sa présence. Le NAR livre au MN les paquets accumulés et détruit le tunnel entre le PAR et le NAR.

[1] Fast Neighbor Advertisement.

Fast Handovers for Hierarchical Mobile IPv6 (F-HMIPv6)

Il existe une amélioration de HMIPv6, basée sur FMIPv6, qui vise à limiter les pertes de paquets lors de la relève (Jung *et al.*, 2005). Cette amélioration vise à obtenir une nouvelle CoA de la part du NAR avant de rompre le lien avec le PAR. Un tunnel bidirectionnel temporaire entre le MAP et le NAR permet de minimiser la perte de paquets pendant la relève.

1.1.7 Résumé des concepts de base

Nous avons vu que le routage conventionnel est basé sur l'adresse destination d'un paquet donné. Nous avons ensuite présenté les types d'applications et le concept de QdS pour les applications temps-réel. Nous avons aussi survolé les concepts de chemins commutés par étiquette et en quoi ils ouvrent la voie à l'ingénierie de trafic. Enfin, nous avons introduit MIPv6 ainsi que les propositions visant à en améliorer les performances.

La révision des concepts de base constitue une entrée en matière pour la définition de la problématique qui est traitée à la section suivante.

1.2 Éléments de la problématique

Nous délimiterons tout d'abord l'étendue du problème que nous aborderons à l'intérieur du réseau d'un fournisseur de services pour usagers mobiles. Ensuite, nous discuterons des considérations particulières issues de la mobilité des usagers.

1.2.1 Délimitation du problème

Les réseaux de fournisseurs de services sont organisés de façon hiérarchique afin de simplifier la gestion de ces derniers. La Figure 1.1 montre les composantes principales de tels réseaux. Ces réseaux se divisent en deux parties principales :

1. le réseau de transport communément appelé dorsale (*backbone*) ;
2. le réseau d'accès à partir duquel les usagers se connectent.

Le réseau de transport

Le réseau de transport a pour but d'acheminer de grands volumes de données entre les différents sous-réseaux du fournisseur de services et l'Internet. En effet, un fournisseur de services pourrait souhaiter diviser les sous-réseaux selon la technologie d'accès employée à l'intérieur de ceux-ci. De même, les critères de division des sous-réseaux pourraient être géographiques.

La gestion de QdS dans un réseau de transport s'effectue au niveau d'agrégats de flots de données dont les caractéristiques sont semblables. Les réseaux de transport gèrent donc un nombre restreint de flots de très grande capacité.

Ceci étant dit, les aspects se rapportant aux réseaux de transport ne seront pas considérés dans le cadre de la recherche de ce mémoire.

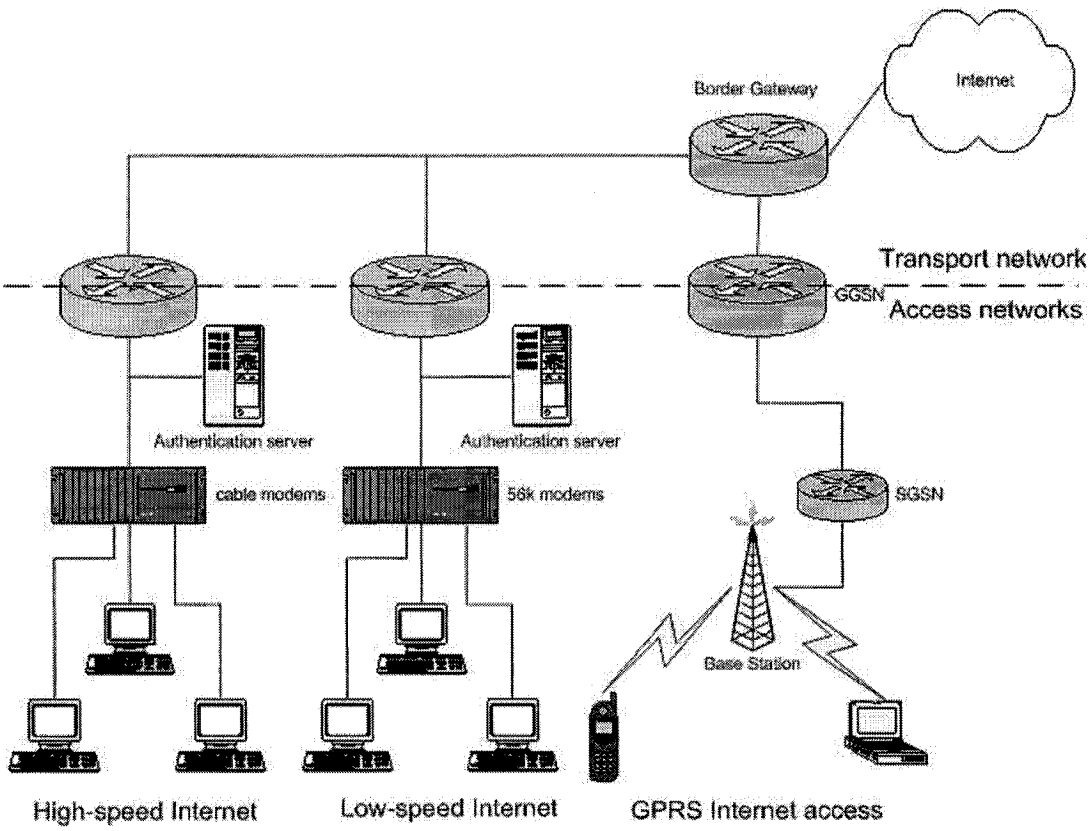

FIGURE 1.1 Composantes principales du réseau d'un fournisseur de services

Le réseau d'accès

Le réseau d'accès a pour mandat de relier les usagers au fournisseur de services par le biais de fils de cuivre, de fibres optiques ou d'ondes électromagnétiques.

Contrairement au réseau de transport, le réseau d'accès doit accomoder un très grand nombre de flots de données de capacité réduite. L'accès à ce réseau est contrôlé par des mécanismes d'authentification.

La gestion de QdS dans un réseau d'accès peut s'effectuer pour chaque flot individuellement ou pour des classes de flots ayant des caractéristiques semblables.

Enfin, dans le cadre de ce mémoire, nous concentrerons tous nos efforts dans le réseau d'accès, plus précisément à partir du routeur d'accès jusqu'au routeur situé à la frontière du réseau de transport. Nous y traiterons du déplacement des réservations de ressources d'un routeur d'accès à un autre. Aussi, nous devrons considérer que les flots de données seront acheminés par des LSP.

1.2.2 Impacts de la mobilité des usagers

Le changement de point d'accès par les usagers mobiles entraîne des problèmes supplémentaires que nous devrons considérer. En effet, la solution proposée devra permettre de déplacer des réservations de ressources ainsi que des LSP lors de la relève. Le processus doit donc être transparent pour l'usager, n'avoir aucune incidence sur la QdS ou entraîner la perte de la connexion.

L'architecture du réseau d'accès

L'architecture du réseau d'accès sera F-HMIPv6. En effet, cette architecture réduit la signalisation intra-fournisseur par rapport au MIPv6 traditionnel. Dans ce mémoire, nous ne traiterons que les relèves intra-fournisseur.

Arkko *et al.* (2003) abordent l'utilisation de IPv6 en réseau cellulaire. Notre proposition de solution devrait s'assurer de ne pas introduire d'incompatibilité inutilement.

Difficultés inhérentes au déplacement des réservations de ressources

Lorsque la relève est initiée, il est primordial de s'assurer que les réservations de ressources pourront être déplacées, sans quoi il y aura dégradation de la QdS et possiblement la perte de la connexion.

Un concept appelé *make-before-break* est appliqué afin que l'on n'effectue la relève que lorsqu'une nouvelle réservation de ressource entre en action et que l'on peut s'assurer de préserver la QdS. L'ancienne réservation de ressource est ensuite détruite.

Il est possible que le déplacement de la réservation de ressource ne puisse s'effectuer par manque de ressources. Dans ce cas, il y a trois avenues possibles :

1. on peut réessayer de nouveau après un délai d'attente ;
2. on peut procéder à la relève sans les réservations de ressources ;
3. on ne fait rien mais on risque la perte de la connexion.

Dans un autre ordre d'idée, lorsque l'on tente de déplacer la réservation de ressource, il est possible que certains routeurs soient communs aux deux chemins parcourus par les flots de données. Il ne faut surtout pas que les routeurs communs comptabilisent en double les demandes de réservations puisqu'il s'agit du même flot de données. Cette double comptabilité pourrait entraîner un rejet de la requête de réservation de ressource alors qu'elle aurait normalement dû être acceptée.

Difficultés inhérentes au déplacement des chemins commutés par étiquette

Les mécanismes actuels de gestion de LSP considèrent que les nœuds source et destination demeurent inchangés tout au long de la session. En contrepartie, seul le parcours composé des nœuds intermédiaires peut changer pour des raisons de panne ou d'ingénierie de trafic.

Dans le cadre de recherche de ce mémoire, le nœud source sera appelé à changer de point d'accès. Il faudra donc élaborer une méthode qui permet d'identifier avec certitude les deux sessions comme étant deux instances d'une seule et même session. Le mécanisme de relève qui sera proposé doit donc créer un nouveau LSP selon le principe *make-before-break*. De plus, il serait souhaitable que les LSP communs aux deux chemins demeurent inchangés lors du processus de relève.

Impacts de la relève sur la QdS

La procédure de relève nécessite l'échange de plusieurs messages visant à déplacer les réservations de ressources, les LSP et possiblement d'autres paramètres définissant la session au niveau de la couche 2 du modèle OSI.

Puisque la relève s'exécute alors que l'usager échange des données par le biais du réseau, il est possible que l'acheminement de certains paquets soit compromis. Bien

entendu, s'il s'agit de trafic *temps-réel*, la QdS sera affectée négativement. Il est donc essentiel d'élaborer un modèle de relève dans lequel cet impact sera minimisé.

La principale difficulté d'un tel modèle est de permettre l'accès à un appareil mobile par deux adresses IPv6 simultanément alors que la relève est en cours.

1.2.3 Résumé des éléments de la problématique

Nous avons tout d'abord identifié le réseau d'accès comme étant le lieu où nous concentrerons nos efforts de recherche. Nous avons ensuite couvert les différents éléments de la problématique liés à la mobilité des usagers. Enfin, nous avons discuté des impacts de la relève sur la QdS rendue à l'usager.

1.3 Objectifs de recherche

Les objectifs de la recherche sont de démontrer que l'on peut intégrer à un réseau d'accès cellulaire basé sur IPv6 les mécanismes permettant l'ingénierie du trafic. Ces mécanismes permettront aux opérateurs de tels réseaux de maximiser l'utilisation de ces derniers tout en étant aptes à satisfaire les requêtes de QdS des usagers mobiles.

Dans un autre ordre d'idée, nous devrons proposer une architecture de réseau permettant de tirer profit du champ `Flow Label` du protocole IPv6 pour créer des LSP. Par contre, l'utilisation de ce champ doit être transparente pour les usagers.

Le besoin de transparence s'explique par le fait que l'on ne veut pas causer d'incompatibilité avec les normes déjà en place. Cela implique que la solution proposée doit restaurer au nœud de sortie (*egress*) la valeur que possédait ce champ en entrant dans le réseau d'accès par le nœud *ingress*.

1.4 Plan du mémoire

Dans le présent chapitre, nous avons survolé les définitions et concepts de base relatifs au sujet de recherche de ce mémoire. Nous y avons aussi présenté les éléments de la problématique et fixé les objectifs de recherche qui devront être atteints.

Le chapitre 2 produira une revue de littérature couvrant les différents aspects de la gestion de QdS dans les réseaux IPv6. Entre autres, nous y présenterons quelques documents importants qui seront à la base de la proposition de solution.

Par ailleurs, les liens entre tous les mécanismes de gestion de QdS ainsi que les architectures de réseaux présentés au chapitre 2 deviendront tangibles au terme de la lecture du chapitre 3. En effet, nous combinerons les idées maîtresses de deux architectures de réseaux ainsi que les mécanismes de réservation de ressources afin de construire des chemins commutés par étiquette à l'aide du champ `Flow Label`.

Ensuite, nous démontrerons dans le chapitre 4 la validation du modèle de la solution qui aura été exposée au chapitre 3. La validation du modèle a pour but d'assurer une couverture adéquate de l'ensemble des conditions d'opération du modèle.

Enfin, le chapitre 5 permettra d'effectuer un retour sur la solution proposée ainsi que sur la validation de cette dernière afin de démontrer qu'elle constitue une avenue de solution possible au problème adressé par le présent mémoire. Nous dresserons une liste des avantages et des inconvénients de la solution proposée. Nous discuterons aussi des limitations entourant les conditions dans lesquelles la solution fut élaborée et présenterons l'état d'avancement des travaux. Enfin, nous aborderons les perspectives de travaux de recherche futurs que le présent mémoire aura soulevées.

CHAPITRE 2

REVUE DE LITTÉRATURE

Nous débuterons ce chapitre par une revue des modèles de réseaux de communication. Ensuite, nous verrons plus en détail les aspects de QdS dans les réseaux IPv6. Dans un autre ordre d'idées, nous présenterons les bases de l'architecture MPLS avant d'aborder les concepts de l'ingénierie de trafic. Enfin, nous résumerons des articles scientifiques pertinents au sujet de recherche de ce mémoire.

2.1 Modèles de réseaux de communication

Cette section constitue un rappel des modèles de commutation dans les réseaux. Nous discuterons aussi des caractéristiques de l'Internet actuel et des considérations liées à l'apparition des applications temps-réel.

2.1.1 Les réseaux à commutation de paquets

Les réseaux à commutation de paquets découpent les messages échangés entre les hôtes en paquets qui sont acheminés individuellement. En effet, chaque décision de routage est indépendante des décisions prises pour les paquets précédents.

Les avantages d'un réseau à commutation de paquets conventionnel sont :

1. un haut degré de survivabilité en cas de panne puisque les paquets peuvent suivre des chemins alternatifs ;
2. aucune ressource autre que la table de routage n'est requise dans les routeurs ;
3. puisqu'un réseau à commutation de paquets est sans connexion, il n'y a aucun délai associé à l'établissement d'une connexion.

Par contre, il y a des inconvénients liés à un réseau à commutation de paquets :

1. il n'y a aucune garantie de service en ce qui a trait à la bande passante, au délai ou à la jigue car toutes les communications partagent un même canal ;

2. il n'y a aucune garantie concernant la livraison ou l'ordre dans lequel les données seront reçues par le destinataire.

Ces inconvénients rendent les réseaux à commutation de paquets conventionnels peu appropriés à l'utilisation d'applications temps-réel. Toutefois, ces lacunes peuvent être compensées par l'établissement de circuits virtuels.

2.1.2 Les réseaux à commutation de circuits

Les réseaux à commutation de circuits fonctionnent selon le modèle du système téléphonique classique où un canal est réservé pendant toute la durée de la connexion. Ces réseaux possèdent plusieurs caractéristiques qui les rendent attrayants pour les applications temps-réel :

1. la bande passante est garantie pendant toute la durée d'une connexion puisque le canal n'est pas partagé ;
2. les paquets sont reçus par le destinataire dans l'ordre qu'ils ont été émis car tous les paquets suivent le même chemin ;
3. les délais de transmission sont fixes et en conséquence la jigue est plus faible que dans les réseaux à commutation de paquets.

Par contre, les réseaux à commutation de circuits ont aussi des inconvénients :

1. il y a un délai pour l'établissement du circuit avant le transfert des données ;
2. il faut maintenir l'état du circuit dans chaque nœud du réseau ;
3. ces réseaux n'offrent aucune protection en cas de panne d'un élément du réseau, ce qui est inacceptable à l'intérieur d'un grand réseau.

2.1.3 Les réseaux de circuits virtuels

Considérant les modèles de commutation de paquets et de circuits, il est clair qu'une solution hybride serait avantageuse. En effet, l'utilisation de circuits virtuels permet de mieux gérer la QdS des connexions tout en assurant la survivabilité du réseau en cas de panne. De plus, les données sont reçues par le destinataire dans l'ordre qu'elles ont été émises à la source. Enfin, le seul inconvénient additionel des circuits virtuels (par rapport à la commutation de circuits) est qu'il y a un délai pour la traduction de l'adresse destination vers l'identificateur de circuit.

2.1.4 Caractéristiques du réseau Internet

Historiquement, le réseau militaire ARPANET, dont découle l'Internet d'aujourd'hui, a été conçu selon le modèle de réseau à commutation de paquets afin de maximiser la survivabilité du réseau en cas d'attaque nucléaire contre le territoire américain (voir Tanenbaum, 2002, pp. 50–54).

En conséquence, le protocole IP a été développé dans le but d'offrir ses services même en cas de congestion ou de perte d'un nœud du réseau. Une décision de routage étant prise pour chaque paquet individuellement, l'adaptation du réseau aux changements des conditions d'opération se fait sans intervention humaine et est transparente pour les usagers. Par contre, le protocole IP n'offre aucune garantie quant à la livraison des paquets à leur destinataire ; la façon dont la transmission des paquets s'effectue est communément appelée *best-effort*.

2.1.5 L'émergence des applications temps-réel

La première proposition de standard du protocole IPv6 (Deering et Hinden, 1995) possède un espace adressable qui dépasse de beaucoup celui qui est disponible pour IPv4. D'ailleurs, on observe une tendance générale dans l'industrie qui pousse le développement de réseaux tout-IP.

Ainsi, chaque appareil, fixe ou mobile, pourrait être doté de plusieurs adresses IPv6. De plus, de nouvelles applications temps-réel font leur apparition régulièrement mais le réseau Internet actuel est encore mal adapté pour en tirer profit.

2.2 Qualité de service dans les réseaux IPv6

Dans cette section, nous aborderons les mécanismes de QdS dans les réseaux IPv6. Nous traiterons d'abord du concept d'identificateur de flot. Ensuite, nous enchaînerons avec la présentation des architectures *DiffServ* et *IntServ* avant d'aborder une proposition visant à combiner les avantages de ces deux architectures. Enfin, nous survolerons le protocole de signalisation RSVP.

2.2.1 L'identificateur de flot dans l'en-tête IPv6

Dans l'en-tête d'un paquet IPv6 tel que décrit par Deering et Hinden (1995), l'identificateur de flot (`Flow Label`) est un champ de 24 bits faisant partie de l'en-tête de base. Son rôle est de permettre à un routeur de reconnaître qu'un paquet doit recevoir un traitement spécial tel qu'une QdS différente de la QdS de base.

Support des flots par les hôtes et les routeurs

Le support des flots est optionnel et actuellement sous expérimentation dans IPv6. Hôtes et routeurs qui ne supportent pas la gestion des flots doivent impérativement émettre tous leurs paquets avec un `Flow Label` à zéro pour indiquer que le paquet n'appartient à aucun flot, ne pas modifier ce champ lors du routage et ignorer ce champ lorsque le paquet leur est destiné.

Identification d'un flot

Un flot de données est identifié par la combinaison de l'adresse IP source et du `Flow Label` (Deering et Hinden, 1995, sect. 6). Plus tard, l'adresse IP destination a été ajoutée à cette combinaison par Rajahalme *et al.* (2004, sect. 1).

Tous les paquets d'un flot doivent obligatoirement être émis avec les mêmes adresses source et destination, `Flow Label`, niveau de priorité et respecter les règles d'utilisation des options IPv6 telles que décrites par Deering et Hinden (1995, sect. 6).

La version actuelle de IPv6 exclut le champ `Traffic Class` de l'identification d'un flot. Ceci permet à un routeur de changer en transit la valeur de ce champ, sans mettre en péril l'intégrité de l'identité du flot (Deering et Hinden, 1998, ann. A).

Avant l'apparition de IPv6, un flot de données était uniquement identifié par les adresses IP source et destination, les numéros de ports (si applicable) et par le protocole de la couche de transport. La définition du `Flow Label` a été modifiée dans la version actuelle du protocole (Deering et Hinden, 1998). En effet, le champ `Prio` de 4 bits a été remplacé par le champ `Traffic Class` qui occupe 8 bits. En conséquence, la taille du champ `Flow Label` est passée de 24 bits à 20 bits (voir la Figure 2.1).

Selon Rajahalme *et al.* (2004), la classification de flots utilisant seulement les adresses IP source et destination ainsi que l'identificateur de flot est efficace puisqu'il devient inutile d'inspecter le contenu du paquet et que les champs mentionnés sont situés à des endroits fixes dans le paquet IPv6.

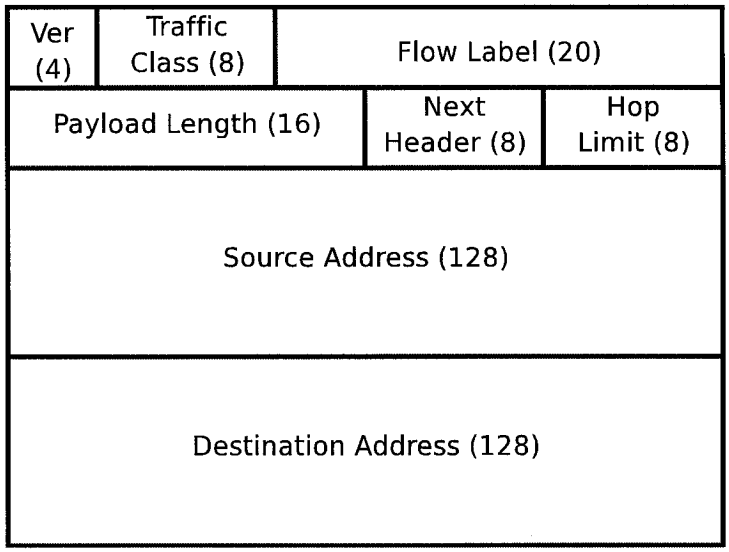

FIGURE 2.1 En-tête d'un paquet IPv6 selon le document RFC2460

Création de flots par opportunisme

Deering et Hinden (1995, sect. 6) réfèrent à la création d'identificateurs de flot de façon opportuniste. Ce mécanisme devait être utilisé lorsque le routeur reçoit un paquet appartenant à un flot pour lequel il ne possède aucune information. Ceci devait aussi permettre à un routeur d'examiner en détail un paquet une seule fois et décider de lui-même quelle serait la QdS à accorder au flot. Toutefois, cette idée a été abandonnée ultérieurement par Deering et Hinden (1998, ann. A).

Considérations relativement à l'utilisation des flots

La spécification de IPv6 laisse en suspens un certain nombre de questions concernant l'utilisation de l'identificateur de flot. Partridge (1995) évoque trois des questions les plus importantes et suggère une façon d'aborder les problèmes qu'elles posent. Les questions évoquées sont présentées ci-dessous.

Q1. Que doit faire un routeur lorsqu'il reçoit un paquet pour lequel il ne possède aucune information sur le flot ? Lors de la période de récupération suivant une panne, il est possible qu'un routeur reçoive des paquets pour lesquels il ne possède aucune information de flot. Il est aussi possible qu'un identificateur de flot

toujours actif mais peu utilisé soit éliminé intentionellement. Dans ces cas, il serait inapproprié de détruire un paquet. La meilleure solution semble de traiter le paquet comme si l'identificateur de flot était nul.

Q2. Comment l'Internet doit-il éliminer les identificateurs de flots ? Il est clair que les routeurs ne doivent pas dépendre de la réception d'un message de libération de flot pour éliminer les informations concernant un flot en particulier. En effet, le message de libération pourrait être perdu avant d'avoir traversé tous les routeurs ou l'appareil ayant réservé un identificateur de flot pourrait tomber en panne.

En conséquence, il faut que les routeurs puissent éliminer d'eux-mêmes les identificateurs de flots inutilisés. La solution est évidemment d'utiliser un compteur qui détruit un identificateur de flot après un délai d'inactivité[1].

Q3. Quels paquets devraient comporter un identificateur de flot non-nul ? Les membres de la communauté IPv6 s'entendent sur le fait qu'une connexion de courte durée ne devrait pas réserver un identificateur de flot. À l'opposé, une connexion temps-réel devrait toujours réserver un identificateur de flot, puisque c'est la raison première de leur création. Par contre, lorsqu'il s'agit de connexions de longue durée qui ne sont pas temps-réel, certains prônent que ces connexions devraient se voir assigner un identificateur de flot, d'autres considèrent que ce serait néfaste.

Un argument en faveur de la création d'un identificateur de flot est que même si la source n'a pas demandé de traitement particulier pour la connexion, le réseau pourrait reconnaître l'application et router le flot de façon à offrir un meilleur service.

Un autre argument favorable est qu'un identificateur de flot peut faciliter le travail de démultiplexage de connexions au niveau des couches réseau et transport du récepteur. Ainsi, un système d'exploitation pourrait plus rapidement traiter un paquet entrant.

À l'inverse, ceux qui s'y opposent affirment que cela aura un impact important sur le fonctionnement des routeurs actuels. En effet, les routeurs sont dotés d'une mémoire cache globale qui mémorise le port de sortie pour chaque destination. La conséquence serait l'explosion de la quantité de mémoire nécessaire pour gérer tous les identificateurs de flots.

[1] Le délai est fixé à 120 secondes par Rajahalme *et al.* (2004, sect. 2).

Résumé de la sous-section

L'identificateur de flot de IPv6 permet de préserver l'indépendance des couches de protocoles en évitant aux routeurs d'inspecter les en-têtes de la couche transport.

Les sections suivantes présenteront deux architectures qui associent une QdS aux différents flots de données, respectivement par classe de trafic et individuellement.

2.2.2 L'architecture *DiffServ*

Une façon simple d'offrir une QdS qui soit évolutive est de regrouper les flots ayant les mêmes caractéristiques en classes de flots qui recevront une même QdS. L'IETF a élaboré une architecture nommée *Differentiated Services* (*DiffServ*) qui est décrite dans les documents de Nichols *et al.* (1998) et de Blake *et al.* (1998).

Le but recherché par l'architecture *DiffServ* est d'offrir un modèle simple de QdS qui peut être rapidement déployé à l'intérieur d'une organisation. Ce modèle a un certain nombre de caractéristiques (Tanenbaum, 2002, pp. 412–413) :

- aucune négociation laborieuse entre les routeurs pour établir un flot ;
- aucune demande de réservation de ressources n'est requise ;
- les routeurs ne stockent aucune information sur les flots de données qui transitent par eux réduisant ainsi les complications en cas de panne d'un routeur ;
- la taille de la mémoire requise sur un routeur est d'autant plus réduite en conséquence de l'item précédent ;
- il peut être facilement implanté sur un routeur vu sa simplicité ;
- tous les flots inclus dans une classe de trafic donnée reçoivent la même QdS ;
- il permet de considérer l'importance relative des flots entre eux.

Les différentes classes de service peuvent se distinguer par les délais de transmission, la jigue et la probabilité de détruire un paquet en cas de congestion. L'architecture *DiffServ* ne spécifie aucun service obligatoire mais définit plutôt les comportements, que l'on nomment *Per-Hop Behaviors* (PHB), adoptés par les routeurs sur tout le chemin parcouru par les paquets (Jha et Hassan, 2002, chap. 7).

Par ailleurs, l'architecture *DiffServ* donne une signification locale aux valeurs prises par l'identificateur de classe de trafic. Toutefois, afin d'identifier de façon unique un PHB lors d'échanges entre deux réseaux distincts, un encodage binaire unifié est suggéré par Black *et al.* (2001). De plus, les traitements effectués par les routeurs lorsqu'un paquet traverse un domaine *DiffServ* et les règles permettant de créer de

tels traitements, appelés *Per-Domain Behaviors* (PDB), sont présentés en détail par Nichols et Carpenter (2001).

Enfin, il est primordial de mentionner que l'un des désavantages de cette classification est que peu importe le nombre de flots qui exigent une certaine QdS, tous partageront les mêmes conditions, ce qui entraîne rapidement une perte de QdS lorsque le nombre de flots augmente.

Le champ Traffic Class dans IPv6

La version de IPv6 décrite par Deering et Hinden (1998), dont le format de l'en-tête est détaillé à la Figure 2.1, offre la capacité de distinguer les flots de données par l'utilisation du champ `Traffic Class` dont l'introduction a nécessité la réduction de la taille du champ `Flow Label` à 20 bits. Toutefois, on souligne la nature expérimentale de l'utilisation de ce champ (Deering et Hinden, 1998, sect. 7).

Nichols *et al.* (1998) définissent le format du champ `DS` (Figure 2.2) qui occupe tout l'espace du champ `Traffic Class` dans l'en-tête d'un paquet IPv6[2]. Les bits 0–5 identifient le *Differentiated Services Code Point* (`DSCP`). Les bits 6–7 identifiés *Currently Unused* ont été réaffectés à la notification explicite de congestion dans le document de Ramakrishnan *et al.* (2001).

Le champ `DSCP` peut prendre 64 valeurs distinctes mais fut divisé en trois plages (Tableau 2.1) afin de faciliter l'assignation de valeurs selon l'usage. Il est important de noter que la valeur 0 indique que le paquet doit recevoir la QdS de base.

Enfin, un certain nombre de clarifications techniques mineures concernant l'usage du champ `DSCP` ainsi qu'une nouvelle terminologie mieux adaptée à l'architecture *DiffServ* sont décrites par Grossman (2002).

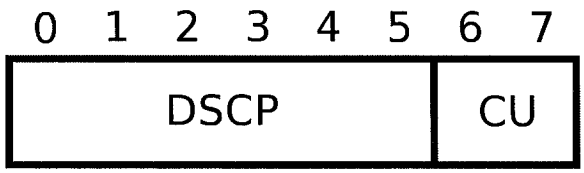

FIGURE 2.2 Format du champ DS tel que défini dans RFC2474

[2]Pour IPv4, le champ `DS` remplace le champ `TOS` tel que défini par Almquist (1992).

TABLEAU 2.1 Plages de valeurs pour le champ DSCP

Plage	Valeurs	Règle d'assignation
1	xxxxx0	Assignation par une norme de l'IANA
2	xxxx11	Expérimentation/Usage local
3	xxxx01	Expérimentation/Usage local (pourrait être jointe à la plage 1)

Le service d'acheminement *Assured Forwarding*

Les documents de Nichols *et al.* (1998) et Blake *et al.* (1998) ont permis le développement de deux principaux services d'acheminement. Le premier service porte le nom de *Assured Forwarding* et est défini par Heinanen *et al.* (1999). Celui-ci a pour but de livrer des paquets en tirant profit des mécanismes de l'architecture *DiffServ*, avec une haute probabilité de réussite, pourvu que le flot de données respecte les proportions de bande passante qui ont été allouées statiquement au niveau de chaque routeur.

Ce groupe de PHB est divisé en quatre classes de service à l'intérieur desquelles un paquet peut se voir assigner l'un des trois niveaux de priorité. Ces niveaux de priorité indiquent quelle est l'importance relative d'un paquet par rapport aux autres à l'intérieur d'une même classe de service. Le Tableau 2.2 montre quels sont les points de contrôle recommandés (Heinanen *et al.*, 1999, sect. 6).

Le réordonnancement des paquets d'un flot est interdit car le service *Assured Forwarding* n'inclut aucune notion de délai ou de jigue associée à l'acheminement.

Le comportement d'une instance de la classe de service *Assured Forwarding* doit réagir à une situation de congestion persistente par l'élimination de paquets et à une situation de congestion temporaire par l'accumulation de paquets en mémoire tampon. Elle doit minimiser la congestion à long terme tout en permettant de courtes périodes de dépassement de capacité.

TABLEAU 2.2 Points de contrôle recommandés du service *Assured Forwarding*

Possibilité d'élimination	Classe 1	Classe 2	Classe 3	Classe 4
Faible	001010	010010	011010	100010
Moyenne	001100	010100	011100	100100
Forte	001110	010110	011110	100110

Le service d'acheminement *Expedited Forwarding*

Le second PHB dont nous allons effectuer le survol porte le nom de *Expedited Forwarding* et la première version de ce service a été définie par Jacobson *et al.* (1999). La version actuelle de ce service est détaillée par Davie *et al.* (2002) et apporte un formalisme mathématique par rapport à la version originale.

Quelques clarifications pour l'implémentation de ce service sont présentées par Charny *et al.* (2002). On y discute des délais à l'intérieur des routeurs et présente des algorithmes qui satisfont les critères du service d'acheminement *Expedited Forwarding.*

Le but de ce PHB est d'offrir un service qui minimise :

- le risque de perte de paquet ;
- le délai de retransmission ;
- la jigue lors de la retransmission.

Le service *Expedited Forwarding* est donc un service d'acheminement prioritaire et des considérations de sécurité supplémentaires sont nécessaires. En effet, les nœuds à la frontière du domaine *DiffServ* doivent contrôler de façon très stricte la bande passante allouée pour ce service et éliminer tout paquet qui dépasse la limite fixée.

La valeur recommandée pour le point de contrôle du service *Expedited Forwarding* est 101110 et provient de la plage 1 définie dans le document à la base de l'architecture *DiffServ* (Nichols *et al.*, 1998, sect. 6).

2.2.3 L'architecture *IntServ*

Contrairement à l'architecture *DiffServ*, l'architecture *IntServ* applique les règles de QdS à chaque flot de données individuellement. Des réservations de bande passante, des garanties de délai et de service peuvent alors être négociées et respectées pendant toute la durée de la connexion.

Les bases de l'architecture *IntServ* sont définies par Braden *et al.* (1994). Le but premier de celle-ci est de tirer profit de la pile de protocoles et des liaisons physiques existantes afin de les utiliser de façon optimale tout en introduisant des services permettant d'assurer l'acheminement de trafic temps-réel.

Support pour le développement de nouveaux services

Le document de Shenker et Wroclawski (1997b) inclut un patron servant à concevoir et évaluer un nouveau service destiné à l'architecture *IntServ.* On y trouve les

définitions et des suggestions pour l'encodage des données servant aux réservations de ressources. Par ailleurs, les paramètres qui constituent les caractéristiques d'un service et les règles de formatage[3] y sont aussi détaillées.

Les composantes de l'architecture *IntServ*

Les composantes de l'architecture *IntServ* que l'on retrouve dans les routeurs sont définies par Braden *et al.* (1994, sect. 2.2) :

1. le *séquenceur de paquets* dont la responsabilité est d'émettre les paquets en ordre de façon à respecter les consignes de QdS ;
2. le *séquenceur de paquets* comprend un sous-composant nommé *estimateur* qui sert à générer des statistiques utilisées par le *séquenceur de paquets* et le *contrôle d'admission* ;
3. le *classificateur de paquets* a la responsabilité de marquer les paquets[4] pour traitement ultérieur par le *séquenceur de paquets* ;
4. le *contrôle d'admission* vérifie qu'une requête de réservation peut être satisfaite sans compromettre les autres réservations en vigueur et authentifie celle-ci ;
5. le *protocole de signalisation* informe les routeurs sur le chemin d'un flot de données que des consignes de QdS doivent être appliquées. Dans le cas d'un réseau IPv6, on utilise généralement RSVP.

Les classes de service supportées par l'architecture *IntServ*

L'architecture *IntServ* doit impérativement supporter les classes de service *best-effort*, *Controlled Link Sharing* et temps-réel. La classe *best-effort* représente la QdS actuellement en vigueur dans l'Internet, sans intervention d'aucun mécanisme de QdS.

La classe de service *Controlled Link Sharing* permet à un opérateur de réseau de diviser les ressources en catégories ayant chacune un pourcentage minimum garanti de la bande passante, en temps de congestion, tout en permettant à une catégorie d'emprunter des ressources aux autres lorsque le lien physique est sous-utilisé.

La classe de service temps-réel se divise en trois modèles de services : le *Guaranteed Service*, le *Controlled Load Service* et le *Null Service.*

[3]Les paramètres sont encodés dans le format *eXtended Data Representation* (XDR, RFC1832).
[4]Le marquage des paquets a pour but d'associer le flot de données à une classe de flots.

Le modèle de service *Guaranteed Service* a pour but d'acheminer des flots de données temps-réel en garantissant la bande passante allouée au trafic ainsi que la borne supérieure de délai de transmission de bout-en-bout. Il offre donc une garantie stricte de service que les nœuds du réseau doivent respecter. Ses propriétés mathématiques sont décrites par Shenker *et al.* (1997).

Le modèle de service *Controlled Load Service* achemine des flots de données à des applications qui tolèrent une variation du délai de transmission et s'adaptent aux conditions changeantes du réseau. Il offre un délai de transmission qui est en général respecté mais qui dépasse parfois la limite spécifiée. Son but est de maximiser l'utilisation des liens physiques tout en offrant un service à coût moindre que le *Guaranteed Service*. Ce service est décrit par Wroclawski (1997b).

Le modèle de service *Null Service* a été développé pour les applications où il est difficile de prévoir les besoins en ressources. Bernet *et al.* (2000b) définissent les spécifications de ce service qui permet à une application de s'identifier à l'agent de QdS en utilisant la signalisation RSVP, et ce sans devoir spécifier ses besoins en ressources. L'agent de QdS applique alors la règle appropriée à l'application qui est définie par l'administrateur du réseau. Ce service est particulièrement utile dans les réseaux supportant à la fois l'architecture *DiffServ* et la signalisation RSVP.

2.2.4 Accès aux services *IntServ* dans les réseaux *DiffServ*

Afin d'offrir une QdS de bout-en-bout, l'architecture *IntServ* doit être supportée par différents types d'éléments de réseau. Dans ce contexte, un réseau *DiffServ* peut être vu comme un élément de réseau faisant partie du chemin complet.

Bernet *et al.* (2000a) proposent de combiner les architectures *IntServ* et *DiffServ* afin d'offrir une QdS de bout-en-bout peu importe la taille du réseau.

L'utilisation de RSVP dans l'architecture *IntServ* effectue des réservations pour des micro-flots au niveau individuel. Au contraire, l'architecture *DiffServ* associe les paquets à des classes de trafic basées sur le champ `DSCP` de l'en-tête du paquet IPv6. Cette dernière peut facilement être implantée sur de très grands réseaux.

Les mécanismes de *IntServ* et *DiffServ* seraient vus comme technologies complémentaires plutôt qu'exclusives. L'architecture *IntServ* garantirait aux applications une QdS de bout-en-bout alors que l'architecture *DiffServ* permettrait l'évolutivité.

Enfin, une alternative qui éviterait la configuration statique des filtres de classification de micro-flots serait d'utiliser RSVP tel que spécifié dans le document de Bernet *et al.* (2000a, sect. 2.3). Le marquage des paquets pourrait être pris en charge par l'hôte avec l'object DCLASS de RSVP (Bernet, 2000) ou par les routeurs si IPSEC n'est pas utilisé[5].

2.2.5 *Resource ReSerVation Protocol* (RSVP)

Le protocole de signalisation utilisé pour la réservation de ressources dans un réseau IPv4 ou IPv6 est RSVP. De nombreux documents complètent la description fonctionnelle de RSVP qui origine de Braden *et al.* (1997).

Il est important de mentionner que RSVP repose sur les protocoles de routage existants pour déterminer le chemin sur lequel les ressources seront réservées. Le rôle de RSVP n'est que d'assurer la QdS sur les chemins choisis par ces derniers.

L'état des réservations est conservé au niveau de chaque routeur et celui-ci doit être rafraîchi régulièrement pour signaler que la réservation est toujours active. Ce mode de fonctionnement permet au réseau de se débarasser éventuellement des réservations qui seraient laissées à elles-mêmes, en cas de défaillance de l'application ayant réservé les ressources ou si le chemin emprunté par le flot de données devait être modifié par les protocoles de routage.

Étant donné qu'un seul agent RSVP existe par nœud du réseau, RSVP circule directement sur IP sans faire appel à un protocole de la couche transport. En effet, il utilise le numéro de protocole 46 mais peut accomoder de vieux systèmes avec UDP.

L'une des caractéristiques importantes de RSVP est que ce protocole manipule les paramètres de QdS et d'admission de façon opaque. En effet, ces paramètres sont passés au module de *contrôle d'admission* pour y être interprétés ; ils ne sont pas utilisés par RSVP directement, mais seulement transportés par ce dernier.

RSVP est un protocole orienté récepteur. En conséquence, le récepteur est responsable de réserver les ressources selon les spécifications de l'émetteur. La raison de cette décision de design découle du fait que RSVP a été développé dans le cadre d'un réseau supportant la large diffusion (*multicasting*) et que les réservations sont fusionnées lorsqu'un routeur détient déjà une réservation pour le flot demandé.

Selon les spécifications de RSVP, un flot de données est défini comme étant uni-

[5]L'encryption de IPSEC masquerait les champs requis pour l'identification des flots.

directionnel. Cela a pour conséquence directe que deux réservations distinctes de ressources sont requises pour une application, telle que la téléphonie, qui génère du trafic bidirectionnel.

Dans la terminologie RSVP, un descripteur de flot (*flow descriptor*) est à la base de toute réservation de ressources. Celui-ci est composé d'une spécification de flot (*flow spec*) et d'une spécification de filtre (*filter spec*) :

- la spécification de flot s'adresse au *séquenceur de paquets* ou à tout autre mécanisme de QdS de la couche 2 de chaque routeur et comprend les attributs qui décrivent les besoins en QdS du flot de données ;
- la spécification de filtre s'adresse au *classificateur de paquets* de chaque routeur et comprend les informations nécessaires à l'identification d'un flot (voir la définition de flot selon Braden *et al.*, 1997, sect. 1.1).

Modèle de réservation

Lors du traitement d'une requête de réservation par un routeur, deux actions principales sont déclenchées :

1. effectuer la réservation de ressources sur les liens sollicités du routeur (incluant le respect des règles de contrôle d'admission dont les extensions à RSVP sont définies par Herzog, 2000) ;
2. transmettre la requête de réservation au prochain routeur en direction de l'émetteur ou groupe d'émetteurs.

Il est possible que la requête de réservation transmise vers le prochain routeur soit différente de celle qui a été initialement reçue. La raison principale est qu'un routeur peut fusionner des réservations en remontant vers la(les) source(s). Lorsqu'une réservation peut être entièrement fusionnée avec une réservation antérieure dans un routeur, la propagation de la requête de réservation se termine à ce routeur.

Styles de réservations

Selon le type d'application, le protocole RSVP peut supporter différents styles de réservations (Braden *et al.*, 1997, sect. 1.3). En effet, il est possible de réserver des ressources pour un flot allant d'un seul émetteur à un seul récepteur ou pour plusieurs émetteurs vers plusieurs récepteurs. Il en résulte que les règles permettant

de fusionner[6] les réservations varient selon le style et ce pour une session donnée.

En conséquence, trois styles de réservations ont été définis dans RSVP selon que la liste des émetteurs soit explicite ou englobante et selon que l'on permette ou non de fusionner les réservations. Le Tableau 2.3 montre les styles de réservations supportés par RSVP.

TABLEAU 2.3 Styles de réservations supportés par RSVP

Sender Selection	Distinct Reservation	Shared Reservation
Explicit	Fixed-Filter (FF) Style	Shared-Explicit (SE) Style
Wildcard	(None defined)	Wildcard-Filter (WF) Style

Le style *Wildcard-Filter* implique que la liste des émetteurs est englobante et que les réservations sont partagées. Dès qu'un nouvel émetteur se manifeste, la réservation s'applique à celui-ci. Lorsqu'une nouvelle réservation est faite, la valeur de la nouvelle réservation est propagée en amont seulement si cette dernière est plus importante que la réservation antérieure pour une même session.

Le style *Fixed-Filter* implique que la liste des émetteurs est explicite et que les réservations sont distinctes. Chaque nouvelle requête entraîne la réservation de ressources supplémentaires.

Le style *Shared-Explicit* implique que la liste des émetteurs est explicite et que les réservations sont partagées. En conséquence, la valeur de la nouvelle réservation est propagée en amont seulement si cette dernière est plus importante que la réservation antérieure pour une même session.

Types de messages dans RSVP

De nombreux messages sont supportés par RSVP. Nous détaillerons les messages les plus importants ci-après. Les messages sont composés d'un en-tête commun suivi d'un ou plusieurs objets de taille variable. La Figure 2.3 montre le format de l'en-tête commun tandis que la Figure 2.4 montre le format des objets de taille variable.

[6]Il ne faut pas oublier que les paramètres des réservations sont opaques à RSVP et doivent donc être traités par des fonctions spécifiques à l'implémentation.

Ver (4)	Flags (4)	Msg Type (8)	Checksum (16)
TTL (8)		Reserved (8)	Length (16)

FIGURE 2.3 Format de l'en-tête commun d'un message RSVP

Length (16)	Class Num (8)	C-Type (8)
Object contents		

FIGURE 2.4 Format d'un objet qui compose un message RSVP

Le message *Resv* permet de réserver des ressources. Ce message origine d'un récepteur et remonte vers le(s) émetteur(s) en suivant l'ordre inverse du flot de données. Chaque routeur doit examiner le message et le propager vers l'amont à moins qu'il y ait fusion complète de la réservation avec une réservation antérieure.

Le message *Path* est créé par un émetteur en direction du(des) récepteur(s) en suivant le chemin emprunté par le flot des données. Ce message comprend entre autres l'adresse du nœud RSVP précédent qui est utilisée dans les messages *Resv*. Cette façon de procéder permet de traverser un nuage de routeurs qui ne supportent pas RSVP. De plus, un message *Path* est composé des éléments suivants :

- le patron d'émetteur (*Sender Template*) qui donne les informations nécessaires à l'identification du flot de données (ces informations utilisent le format d'une spécification de filtre et sont utilisées par un *classificateur de paquets* pour marquer les paquets) ;
- les caractéristiques du trafic généré par l'émetteur qui sont utilisées par le module de contrôle du trafic d'un routeur pour éviter la sur-réservation, et vraisemblablement des erreurs au niveau du *contrôle d'admission* (*Sender Tspec*) ;
- des informations optionnelles[7] mises à jour à chaque routeur qui permettent à une application de mieux déterminer la QdS de bout-en-bout (*Adspec*).

[7]Ces informations utilisent le mécanisme *One Pass With Advertizing* (OPWA) qui est une amélioration du mécanisme usuel à une passe.

Les messages *PathTear* et *ResvTear* servent à annuler une réservation ou un chemin de données. On utilise l'un ou l'autre de ces messages selon que la demande d'annulation de réservation origine de l'émetteur ou du récepteur. Le message est propagé à tous les routeurs impliqués.

L'utilisation des messages *PathTear* et *ResvTear* est facultative, puisque les routeurs élimineront éventuellement la réservation, mais elle est fortement suggérée car cela permet de libérer plus rapidement les ressources pour d'autres usages.

Le message de confirmation *ResvConf* permet, si la requête de réservation émise contient un objet de type *Confirmation-Request*, d'indiquer à l'application que la réservation a été effectuée avec succès.

À chaque point de fusion de spécifications de flot, seule la plus grande requête ainsi que l'objet *Confirmation-Request* sont acheminés vers l'émetteur. Lorsque la fusion est complète ou si la réservation est acceptée par l'émetteur, un message *ResvConf* est retourné vers l'application qui a effectué la réservation.

Il y a deux conséquences au mécanisme de confirmation décrit précédemment :

- la réception par un routeur d'une requête de ressources plus grande que celle déjà en place entraîne l'émission d'un message d'erreur ou de confirmation ;
- la réception d'un message *ResvConf* n'offre aucune garantie que la réservation soit effective (un exemple est donné par Braden *et al.*, 1997, sect. 2.6).

Le message d'erreur *ResvErr* peut être émis par un routeur ayant refusé une requête de réservation dont la syntaxe est pourtant correcte. Un nœud du réseau peut aussi décider unilatéralement de terminer la réservation.

Étant donné que l'échec d'une requête de réservation peut avoir pour origine la fusion de réservations antérieures, un message *ResvErr* doit être envoyé à tous les récepteurs concernés.

De plus, la fusion de réservations pose deux problèmes, couramment appelés *killer reservation problems* :

1. une requête de réservation plus importante que celle déjà en place qui est refusée par un routeur en amont ne doit pas affecter la réservation antérieure ;
2. une application qui persiste à effectuer une requête de réservation qui se voit refusée en amont ne doit pas empêcher une autre réservation, plus petite et qui serait normalement acceptée, de se produire.

RSVP dans l'architecture *IntServ* de l'IETF

Le document de Wroclawski (1997a) détaille l'utilisation de RSVP dans le cadre de l'architecture *IntServ* définie par l'IETF. Ce dernier décrit l'usage des modèles de service *Guaranteed Service* ainsi que le *Controlled Load Service* en définissant le format d'objets permettant à RSVP de transporter les informations nécessaires à l'établissement de réservations de ressources. On y retrouve entre autres le format des objets `FLOWSPEC`, `ADSPEC` et `SENDER_TSPEC`.

L'objet `FLOWSPEC` décrit les paramètres de QdS demandés par un récepteur. Celui-ci peut être modifié à un point de fusion si nécessaire.

L'objet `ADSPEC` définit les propriétés du chemin, les paramètres requis pour réserver des ressources et la disponibilité des services. Ce message origine de la source, ou d'un nœud intermédiaire, vers les récepteurs et peut être modifié en cours de route pour refléter l'état des liens. Il permet aussi de déterminer le MTU du chemin.

L'objet `SENDER_TSPEC` décrit les caractéristiques du trafic généré par la source et ne peut être modifié par aucun nœud du réseau.

Problèmes reliés à l'utilisation de RSVP et conclusions

L'introduction de RSVP augmente considérablement le niveau de complexité des routeurs par l'ajout des composants nécessaires à l'architecture *IntServ*.

De plus, puisque RSVP est orienté récepteur, l'évolutivité de RSVP est excellente lorsque le nombre de réservations par flot augmente mais elle est $O(n)$ lorsque les réservations sont distinctes (Jha et Hassan, 2002, sect. 6.7).

Nous avons aussi fait mention des *killer reservation problems* qui doivent être évités lors des fusions de requêtes de réservation de ressources.

Enfin, RSVP ne répond pas à la question : qui peut réserver des ressources ? L'IETF a mis sur pied un groupe de travail qui développe un protocole permettant de répondre à ce besoin, nommé *Resource Allocation Protocol* (RAP).

2.3 L'architecture MPLS

Le réseau cœur de l'Internet est doté d'une architecture de transport basée sur les chemins commutés par étiquette qui coexiste avec le modèle de routage classique basé sur l'adresse destination tout en étant plus performante que celui-ci.

Les fondements de l'architecture *MultiProtocol Label Switching* (MPLS) ont été élaborés par Rosen *et al.* (2001b). Cette architecture a été initialement développée par un groupe d'ingénieurs de Cisco et portait alors le nom de *Tag Switching* avant d'être renommée *Label Switching* lorsqu'elle a été proposée à l'IETF pour mener à sa standardisation. Le qualificatif *MultiProtocol* vient du fait que n'importe quel type de protocole de la couche réseau peut être acheminé par MPLS.

2.3.1 Acheminement de paquets dans un réseau MPLS

Nous avons mentionné que dans un réseau IP, et par extension pour tous les protocoles de couche réseau qui ne sont pas orientés connexion, une décision de routage est prise, dans chaque routeur, pour chaque paquet individuellement.

De plus, un en-tête de paquet contient beaucoup plus d'informations que le minimum requis afin de choisir le saut vers le prochain routeur. Le processus d'acheminement se compose de deux fonctions essentielles :

1. déterminer la classe d'acheminement (FEC) du paquet ;
2. associer chaque classe d'acheminement à un routeur destination.

Pour l'acheminement conventionnel de paquets IP, un routeur considère que deux paquets quelconques font partie de la même FEC s'ils réfèrent à la même entrée dans la table de routage.

Dans le cas de MPLS, l'association du paquet à une FEC est effectué une seule fois, soit à l'entrée du réseau. On encode la FEC choisie dans une étiquette qui sert à l'acheminement, évitant ainsi aux routeurs intermédiaires d'analyser l'en-tête de la couche réseau.

Lorsqu'un routeur MPLS, communément appelé *Label Switching Router* (LSR), reçoit un paquet, ce dernier utilise l'étiquette du paquet reçu en tant qu'index dans une table qui spécifie le prochain routeur, la FEC et une nouvelle étiquette qui remplacera l'étiquette actuelle, avant de transmettre le paquet vers le prochain routeur.

La méthode d'acheminement proposée par l'architecture MPLS comporte de nombreux avantages :

- le processus d'acheminement peut être effectué par des appareils qui ne peuvent analyser l'en-tête du protocole de la couche réseau ou qui ne pourraient pas le faire à une vitesse adéquate ;
- l'association d'une FEC à l'entrée du réseau permet au routeur d'accès de tenir compte d'informations qui ne sont pas contenues dans l'en-tête du paquet ;
- un paquet entrant dans le réseau, par un routeur en particulier, pourrait se faire attribuer une étiquette différente de celle d'un paquet identique entrant par un autre routeur (ceci est impossible avec l'acheminement conventionnel) ;
- le degré de complexité de la tâche d'associer un paquet à une FEC est indépendant de la capacité des routeurs intermédiaires à les acheminer ;
- il est parfois souhaitable de déterminer une route, plutôt que d'utiliser les mécanismes de routage dynamique, avant même qu'un paquet entre dans le réseau pour des raisons administratives ou d'ingénierie de trafic.

2.3.2 Format de l'en-tête MPLS

La Figure 2.5 montre le format de l'en-tête MPLS tel que détaillé dans le document de Rosen *et al.* (2001a). Celui-ci est composé de 32 bits et comprend les éléments suivants :

- le champ `Label` contient l'étiquette[8] ;
- le champ `Exp` est réservé pour usage expérimental ;
- le bit `S` prend la valeur 1 lorsque le bas de la pile d'étiquettes est atteint ;
- le champ `TTL` (*Time-To-Live*) permet l'élimination éventuelle de la trame en cas de boucle dans le réseau.

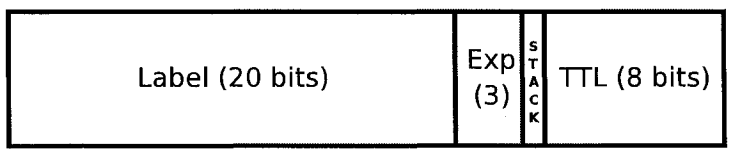

FIGURE 2.5 En-tête d'une trame MPLS

[8]Notez que la taille de l'étiquette MPLS est identique à celle d'un identificateur de flot en IPv6.

2.3.3 Comparaison avec ATM

Les architectures MPLS et ATM ont plusieurs points en commun :

- toutes deux sont des architectures de transport et en conséquence ne sont pas composées d'hôtes ;
- elles associent un flot à une FEC (on utilise une étiquette sous MPLS alors qu'ATM combine le VCI et le VPI) ;
- les nœuds en périphérie du réseau sont les seuls à effectuer du routage, les nœuds intermédiaires ne font que de l'acheminement ;
- elles peuvent transporter divers types de protocoles de la couche réseau.

Par contre, MPLS élimine le besoin de diviser les flots en cellules[9] puisque la taille des paquets ne pose plus un problème de délai vu l'immense capacité des liens optiques actuels. Ce problème était une motivation du développement de l'architecture ATM.

2.3.4 Assignation et propagation des étiquettes

Dans un réseau MPLS, la décision d'associer une étiquette à une FEC en particulier revient au nœud qui est en aval par rapport au lien où l'association prend force. La distribution des étiquettes se fait donc des nœuds en aval vers les nœuds en amont.

2.3.5 Modèles de distribution des étiquettes

L'architecture MPLS définit deux modèles de distribution d'étiquettes. Certaines implémentations de MPLS pourront supporter l'un ou l'autre de ces modèles de distribution qui sont décrits dans les paragraphes suivants :

Le modèle de distribution *Downstream-on-demand* permet à un LSR de demander explicitement à un LSR en aval, pour une FEC en particulier, la valeur de l'étiquette associée à cette FEC.

Le modèle de distribution *Unsollicited Downstream* permet à un LSR donné de transmettre à ses voisins les associations aux FEC sans requête explicite.

[9]Une cellule ATM est composée 53 octets, dont 5 octets pour l'en-tête.

2.3.6 La pile des étiquettes MPLS

Alors qu'une seule étiquette est requise pour traverser un réseau MPLS, une approche plus générale est de considérer un ensemble d'étiquettes organisé en une pile. Le traitement d'une étiquette est complètement indépendant du niveau où cette dernière se trouve (Rosen *et al.*, 2001a).

Les opérations supportées par la pile sont les suivantes :

- remplacer l'étiquette à la tête de la pile par une nouvelle étiquette ;
- retirer l'étiquette à la tête de la pile ;
- remplacer l'étiquette à la tête de la pile par une nouvelle étiquette, puis insérer une ou plusieurs autres étiquettes à la tête de la pile.

2.3.7 Les chemins commutés par étiquette

Les chemins commutés par étiquette, dont le terme anglais est *Label-Switched Path* (LSP) sont à la base de l'architecture MPLS. Le LSP pour un paquet donné est constitué d'une séquence de routeurs dont :

- le premier LSR ("LSP Ingress") qui ajoute une étiquette à la pile d'étiquettes ;
- tous les routeurs intermédiaires qui basent leur décision d'acheminement selon l'étiquette à la tête de la pile ;
- le dernier LSR ("LSP Egress") qui retire l'étiquette à la tête de la liste et achemine le paquet selon une étiquette MPLS à un niveau inférieur ou en utilisant les mécanismes d'acheminement du protocole de la couche réseau si la dernière étiquette a été retirée.

En conséquence, un LSP peut être un sous-ensemble de tous les LSR qui seront traversés par un paquet donné et porte alors le nom de *Tunnel*. Il est aussi important de mentionner que l'architecture MPLS permet l'agrégation de FEC afin de réduire le nombre d'étiquettes à gérer ainsi que le trafic de distribution d'étiquettes.

2.3.8 Protocoles de distribution d'étiquettes MPLS

L'architecture MPLS offre la liberté de choisir le protocole de distribution des étiquettes selon Rosen *et al.* (2001b, sect. 3.6). En effet, au moment de la parution de ce document, plusieurs protocoles existants avaient des extensions en voie de standardisation.

2.3.9 Sélection des routes

L'architecture MPLS supporte deux modes de sélection de routes :

1. le mode *Hop-by-hop* où la décision d'associer une FEC à un port de sortie revient à chaque LSR ;
2. le routage explicite qui peut être utilisé pour la mise en place de règles de routage ou d'ingénierie de trafic.

2.3.10 Tunnels LSP et réservations de ressources

Il est possible de combiner les étiquettes MPLS avec une requête RSVP de réservation de ressources. En effet, cette approche permet d'assigner une étiquette MPLS à une spécification de flot RSVP, évitant ainsi d'inspecter les champs du paquet des couches réseau et transport à chaque LSR. De plus, ce protocole de signalisation peut être utilisé pour distribuer les étiquettes.

2.3.11 Résumé de l'architecture MPLS

Nous avons vu que MPLS est basée sur les chemins commutés par étiquettes ainsi que les avantages de cette approche. Enfin, nous avons discuté de la propagation des étiquettes pour faire le lien avec la section suivante qui traitera d'ingénierie de trafic.

2.4 Ingénierie de trafic

L'ingénierie de trafic est essentielle à la gestion des réseaux modernes car elle permet l'atteinte d'objectifs visant à rentabiliser les équipements très coûteux tout en offrant une QdS aux utilisateurs. Cette discipline est apparue en raison de la nature commerciale et compétitive de l'Internet. D'ailleurs, une proposition permettant aux fournisseurs de services d'offrir la QdS par agrégation de flots, tirant profit de l'architecture MPLS, de RSVP, ainsi que l'ingénierie de trafic est élaborée dans le document de Li et Rekhter (1998).

2.4.1 Objectifs de l'ingénierie de trafic

Les objectifs cruciaux de l'ingénierie de trafic peuvent être classés selon qu'ils visent l'utilisation des ressources ou le contrôle du trafic.

Les objectifs orientés sur le trafic

Ces objectifs incluent les aspects qui améliorent la QdS des flots de données. Dans un réseau ayant une seule classe *best-effort*, les objectifs[10] peuvent être de :

- minimiser la perte de paquets ;
- maximiser l'utilisation de la bande passante ;
- minimiser le délai de transmission ;
- assurer le respect des ententes de service, communément appelées *Service Level Agreements* (SLA).

Les objectifs orientés sur les ressources

Ces objectifs visent à équilibrer l'utilisation des chemins alternatifs en diminuant la congestion ou en évitant la sous-utilisation de certains nœuds.

Les problèmes de congestion causés par une gestion inefficace des ressources peuvent être réglés par l'ingénierie de trafic. En effet, on peut introduire des règles de balancement de trafic qui améliorent significativement la QdS offerte par le réseau.

2.4.2 Le processus d'ingénierie de trafic

Selon Awduche *et al.* (1999, sect. 2.2), on peut considérer un réseau comme étant un système avec boucle de rétroaction pour lequel l'ingénieur de trafic, ou un automate adapté à cette tâche, joue le rôle de contrôleur. Les actions effectuées par l'ingénieur de trafic sont les suivantes :

1. formulation d'une règle de contrôle ;
2. observation de l'évolution de l'état du réseau ;
3. caractérisation du trafic ;
4. application d'actions correctrices dans le but d'amener le réseau dans un certain état qui est en accord avec la règle de contrôle.

2.4.3 Ingénierie de trafic dans les réseaux MPLS

Awduche *et al.* (2002) traitent des grands principes de l'ingénierie de trafic intra-domaine dans l'Internet en mettant l'emphase sur l'utilisation de MPLS.

[10] On remarquera que la définition des bits du champ `TOS` de IPv4 répond aux trois premiers objectifs de la liste (voir Almquist, 1992).

Les raisons qui rendent attrayante l'utilisation de MPLS dans le cadre de l'ingénierie de trafic sont (Awduche *et al.*, 1999, sect. 3.0) :

- l'utilisation de LSP, dont l'acheminement est indépendant de l'adresse destination et dont la gestion et la création peuvent être facilement automatisées ;
- l'association de flots de trafic à un LSP ;
- l'influence d'attributs sur le comportement des flots de données ;
- la capacité d'agrégation et de désagrégation de flots, alors que le modèle classique d'acheminement basé sur l'adresse destination ne permet que l'agrégation ;
- facilité d'implémentation d'un modèle de routage basé sur les contraintes ;
- délai de traitement inférieur à de nombreuses alternatives pour l'ingénierie de trafic vu sa simplicité ;
- par l'utilisation des LSP, implémentation d'un modèle de réseau à commutation de circuits reposant sur le modèle de routage actuel de l'Internet.

Les graphes MPLS induits

Le concept de graphe MPLS induit est primordial pour l'ingénierie de trafic dans les réseaux MPLS (Awduche *et al.*, 1999, sect. 3.1). En effet, ce dernier est logiquement associé au réseau physique par la sélection de LSP qui regroupent les flots de données.

Un graphe MPLS induit consiste en un ensemble de LSR représentés par les nœuds du graphe et un ensemble de LSP qui composent les arcs. Il est possible de créer des graphes hiérarchiques grâce à la pile d'étiquettes MPLS.

Leur importance découle de la difficulté liée à la gestion de la bande passante qui consiste à appliquer le graphe MPLS induit à la topologie physique du réseau.

2.4.4 *RSVP for Traffic Engineering* (RSVP-TE)

Le protocole RSVP a été étendu pour supporter la création de LSP dans le cadre de l'architecture MPLS[11] ; ces extensions sont consignées par Awduche *et al.* (2001a). RSVP-TE comprend de nouveaux objets RSVP dont le but est d'établir des LSP routés explicitement, utilisant RSVP comme protocole de signalisation. Nous ferons dans cette sous-section un survol des caractéristiques de RSVP-TE ainsi que des objets qui ont été ajoutés pour les échanges entre les routeurs.

[11] RSVP-TE s'applique aussi aux LSP créés sous les architectures ATM et Frame Relay mais nous n'en tiendrons pas compte dans ce mémoire.

Les tunnels LSP pour l'ingénierie de trafic

Awduche *et al.* (2001a) introduisent le concept de nœud abstrait. Un nœud abstrait est un groupe de nœuds dont la topologie est opaque au nœud *ingress* d'un LSP. Un nœud abstrait est dit simple si celui-ci est composé d'un seul nœud physique.

Puisqu'un flot à l'intérieur d'un LSP est uniquement identifié par l'étiquette MPLS appliquée au LSP par le nœud *ingress* de ce dernier, ces LSP peuvent être traités comme des tunnels entre des nœuds abstraits.

Une application clé de ces tunnels est l'ingénierie de trafic. Awduche *et al.* (1999) expliquent quelles sont les motivations qui ont mené au développement de RSVP-TE.

Les tunnels LSP permettent la création d'un ensemble de règles visant l'optimisation des performances du réseau. Par exemple, les tunnels LSP peuvent être routés manuellement ou automatiquement pour contourner un point de panne ou de congestion. De plus, plusieurs LSP peuvent être créés entre deux nœuds et le trafic peut ainsi être associé à l'un des tunnels selon les règles locales en place.

Un avantage à utiliser RSVP-TE pour l'établissement de LSP est que celui-ci peut optionnellement réserver des ressources en cours de route.

Objets spécifiques à RSVP-TE

Avant d'aborder la création de LSP à l'aide de RSVP-TE, nous devons présenter les objets spécifiques à RSVP-TE ainsi que leur utilisation. En plus des nouveaux objets décrits ci-dessous, RSVP-TE regroupe les objets RSVP suivants sous la banière générique `LSP_TUNNEL_IPV{4,6}` : `RSVP_SESSION`, `SENDER_TEMPLATE` et `FILTER_SPEC`.

L'objet `EXPLICIT_ROUTE` comprend la liste des nœuds qui doivent être traversés[12] afin de permettre le routage explicite dont le but est de maximiser les performances du réseau. Cet objet est inclus au besoin dans le message *Path* généré par l'émetteur. Le routage explicite peut être *strict* ou *souple* selon que la liste des nœuds à traverser doive être *rigoureusement respectée* ou seulement *suggérée*.

L'objet `LABEL_REQUEST` est inclus dans le message *Path* de l'émetteur afin de demander l'attribution d'une étiquette. Il indique aussi le type de protocole de la couche réseau qui circule dans le LSP en instance de création.

[12]Le concept de nœud abstrait permet de généraliser le routage explicite en une suite de préfixes IP ou de systèmes autonomes (AS).

L'objet `LABEL` est retourné par le nœud *egress* à l'intérieur d'un message *Resv* et indique au nœud précédent l'étiquette assignée pour un LSP donné. Chaque routeur doit modifier la valeur de l'étiquette de l'objet `LABEL` pour identifier sa propre FEC qui est associée au LSP.

Création de LSP à l'aide de RSVP-TE

La création d'un LSP avec RSVP-TE se fait en deux étapes en utilisant la méthode *Downstream-on-demand* pour la propagation d'étiquettes de l'architecture MPLS :

1. l'émetteur envoie un message *Path* avec `LSP_TUNNEL_IPV{4,6}` comme identificateur de session, augmenté d'un objet `LABEL_REQUEST` et optionnellement d'un objet `EXPLICIT_ROUTE` ;
2. le récepteur répond par un message *Resv* et propage l'étiquette choisie vers l'émetteur dans un objet `LABEL`.

Reroutage des LSP pour des fins d'ingénierie de trafic

Le protocole RSVP-TE permet le reroutage d'un LSP pour des fins d'ingénierie de trafic. D'ailleurs, afin de minimiser le risque de perturbation pour un LSP, on utilise une méthode couramment appelée "*make-before-break*".

Cette méthode s'assure que le LSP est correctement rerouté avant d'entreprendre la libération des ressources utilisées par le LSP actuel. En conséquence, il faut que les portions de chemin qui sont communes à l'ancien et au nouveau LSP ne voient pas leurs ressources comptabilisées en double par les routeurs concernés[13].

Reroutage des LSP pour contourner un point de panne

Le reroutage des LSP peut aussi être effectué pour contourner un point de panne (lien ou nœud). Pan *et al.* (2005) ont élaboré deux méthodes de protection de LSP activées par deux objets supplémentaires contenus dans le message `PATH`.

L'objet `DETOUR` est utilisé pour protéger un seul LSP et spécifie les LSP de protection sous la forme d'une liste d'adresses IPv4 ou IPv6.

[13] Un identificateur nommé `TUNNEL ID` permet de reconnaître un même tunnel circulant sur deux LSP différents.

L'objet `FAST_REROUTE` spécifie le lien à utiliser en cas de panne et un ensemble de paramètres dont les filtres d'attributs de sessions et la bande passante à utiliser comme protection. Cette méthode de protection est appliquée à ensemble de LSP.

Revue de RSVP-TE

Nous avons survolé les principales caractéristiques de RSVP-TE dans le cadre de l'ingénierie de trafic. Même si RSVP-TE peut être utilisé pour les architectures ATM et Frame Relay, nous avons restreint la revue de RSVP-TE à l'architecture MPLS.

Awduche *et al.* (2001b) couvrent les aspects d'applicabilité de RSVP-TE et rappelle que les deux principales différences entre RSVP et RSVP-TE sont que RSVP agit sur des micro-flots alors que RSVP-TE agit sur des LSP. De plus, le protocole RSVP-TE généralise le concept de flot élaboré dans RSVP.

Enfin, il est important de mentionner les limitations de RSVP-TE. En effet, RSVP-TE ne supporte que les LSP point-à-point et unidirectionnels. De plus, les états des LSP doivent être régulièrement rafraîchis, et ce de par la nature de RSVP.

2.4.5 Conclusions pour l'ingénierie de trafic

L'ingénierie de trafic est maintenant une discipline essentielle à la rentabilité d'un réseau en permettant de maximiser l'utilisation des ressources de ce dernier. Tout d'abord, nous avons présenté les objectifs de l'ingénierie de trafic. Ensuite, nous avons traité de l'ingénierie de trafic dans les réseaux MPLS. Enfin, nous nous sommes attardés à RSVP-TE qui est une extension de RSVP pour les LSP.

2.5 Mobilité et réservations de ressources

La mobilité des usagers pose de grandes difficultés lorsqu'il s'agit de garantir la QdS des connexions temps-réel. En effet, il est essentiel que les réservations de ressources suivent les déplacements de l'usager, d'un point d'accès à un autre, sans dégrader significativement la QdS et en évitant les risques de perte de connexion liés à la rareté des ressources.

Des propositions de solutions visant à résoudre ces problèmes ont été abordés dans la littérature scientifique. Nous présenterons les solutions jugées les plus importantes dans le cadre de ce mémoire.

2.5.1 Le protocole FH-RSVP

Le but du protocole *Fast Handover Resource Reservation Protocol* (FH-RSVP) élaboré par Elleingand (2004) est de tirer profit du fait que la micromobilité représente la grande majorité des déplacements des usagers. Nous présenterons les caractéristiques principales de ce protocole qui sont pertinentes à notre sujet de recherche.

Réservations à l'intérieur du réseau d'accès

L'idée avancée par Elleingand implique que les réservations de ressources se limitent au réseau d'accès, c'est-à-dire entre le *Access Edge Site* (AES) et le *Border Edge Site* (BES). Le BES est responsable d'effectuer une réservation de ressources sur le lien descendant, au nom du CN, et ce pour des raisons de sécurité.

En conséquence, les réservations de ressources entre les *systèmes autonomes* (AS) seraient effectuées dans le cadre d'ententes de service entre les fournisseurs.

Caractéristiques de FH-RSVP

La sémantique des messages de ce protocole est fortement inspirée de RSVP mais, à l'opposé de RSVP qui est orienté récepteur, FH-RSVP est orienté émetteur.

Par ailleurs, FH-RSVP se limite aux sessions point-à-point alors que RSVP permet de réserver des ressources dans le cadre de sessions à large diffusion.

Dans le cas où deux interlocuteurs se trouveraient dans le même réseau d'accès, la connexion s'effectue directement, sans passer par un nœud central.

En raison de la mobilité du MN, Elleingand propose d'utiliser l'adresse HoA du MN dans l'objet `SESSION` qui est utilisé par les routeurs pour identifier la session.

Enfin, l'implémentation de FH-RSVP est basée sur l'architecture hiérarchique de HMIPv6 et F-HMIPv6 pour la gestion de la relève intra-domaine.

2.5.2 Le protocole HPMRSVP

Dans une proposition qui partage une base commune avec FH-RSVP, Abondo et Pierre (2004) proposent le protocole *Hierarchical Proxy Mobile Resource Reservation Protocol* (HPMRSVP) qui ajoute de nombreux raffinements par rapport à FH-RSVP.

Caractéristiques de HPMRSVP

Le protocole HPMRSVP possède toutes les caractéristiques de FH-RSVP mais modifie le traitement des messages de rafraîchissement.

En effet, les messages de rafraîchissement d'états de chemin et de réservation sont échangés entre le AES et le BES sans impliquer le MN. En conséquence, HPMRSVP préserve les ressources radio.

Modification d'une réservation de ressources

Alors que la réservation initiale des ressources s'effectue par l'échange de messages limités aux réseaux d'accès des usagers, la modification d'une réservation implique l'échange de messages de bout-en-bout. Toutefois, cette signalisation est complexe et comporte des cas spéciaux qui rendent difficile l'implémentation de ce protocole en matériel.

2.5.3 Le protocole MRSVP

Talukdar *et al.* (2001) proposent une extension à l'architecture *IntServ* qui permet à un MN d'effectuer des réservations de ressources, par anticipation, dans les zones où il est possible qu'il migre. Ainsi, les réservations sont effectuées à partir d'un ensemble de zones, appelé *Mobile Specification* (MSPEC).

Afin de maximiser l'utilisation des ressources, une réservation active est effectuée dans la zone courante du mobile tandis que les autres réservations sont passives. Une réservation passive peut être partagée avec d'autres MN pour le transport de paquets de moindre importance tandis qu'une réservation active est exclusive.

L'avantage de MRSVP réside dans une réduction importante du délai de rétablissement des paramètres de QdS lors de la relève.

En contrepartie, MRSVP modifie significativement le protocole RSVP et augmente considérablement la charge de traitement des AR avoisinants qui maintiennent des états de réservations additionnels. De plus, pour atteindre ses objectifs, MRSVP gaspille des ressources sur les AR voisins en les réservant dans l'espoir de les utiliser éventuellement. Enfin, Talukdar *et al.* n'expliquent pas le mécanisme par lequel un MN peut obtenir sa MSPEC.

2.5.4 Le protocole HMRSVP

Tseng *et al.* (2003) proposent une version améliorée de MRSVP (Talukdar *et al.*, 2001) qui réduit le nombre excessif de réservations passives de MRSVP en n'effectuant les réservations passives qu'à l'intérieur des cellules frontières entre deux régions distinctes. HMRSVP combine les idées maîtresses de MRSVP tout en constituant une hiérarchie de routeurs (qui possède une ressemblance frappante avec HMIPv6), ce qui évite la propagation de bout-en-bout des messages des réservations passives.

Alors que HMRSVP réduit considérablement le nombre de réservations passives, comparativement à MRSVP, celui-ci conserve le haut degré de complexité du protocole dont il s'inspire.

2.6 Architecture à commutation d'étiquettes dans les réseaux IPv6

Cette section effectue un survol de deux propositions d'architectures de transport qui permettent l'acheminement de paquets basé sur la commutation d'étiquettes. Ces architectures ne proposent toutefois aucun mécanisme de distribution des étiquettes.

2.6.1 *IPv6 Label Switching Architecture* (6LSA)

Une approche combinant les identificateurs de flot IPv6 ainsi que les chemins commutés par étiquette est proposée par Chakravorty (2005).

En effet, cette approche consiste à utiliser le champ `Flow Label` comme étiquette qui spécifie la FEC du paquet, de façon similaire à MPLS. Chaque routeur devra considérer la valeur du champ `Flow Label` ainsi que l'interface d'entrée du paquet plutôt qu'uniquement l'adresse IP destination de ce dernier.

Flots de données dans l'architecture 6LSA

Dans l'architecture 6LSA, le champ `Flow Label` d'un paquet identifie aussi bien le flot de données que la FEC à laquelle il appartient. De plus, cette valeur n'a qu'une signification locale, à moins que la valeur soit partagée par plusieurs nœuds.

Un flot est donc une séquence de paquets en provenance d'un nœud source et peut être envoyé à tout type d'adresse IP destination. Le nœud source demande un

traitement particulier pour le flot de données qui peut être spécifié aux routeurs par un protocole de signalisation, par des mécanismes de l'architecture *DiffServ* ou par des informations disponibles à même le flot de données. Enfin, un flot est identifié par la valeur du champ `Flow Label` ainsi que les adresses source et destination.

Caractéristiques de l'architecture 6LSA

Les caractéristiques de l'architecture 6LSA proposée par Chakravorty sont :

- 6LSA offre un acheminement rapide de paquets IPv6 grâce à l'identificateur de flot et permet l'utilisation de mécanismes de QdS ou services de livraison ;
- il est possible de configurer un réseau 6LSA de façon à éviter la nécessité d'implanter un mécanisme de distribution d'étiquettes ;
- la combinaison des champs `Flow Label` et `Traffic Class` offre 2^{28} valeurs possibles pour la différenciation de classes de trafic et de la QdS ;
- toutes les options IPv6 sont utilisables dans l'architecture 6LSA ;
- 6LSA n'agit qu'au niveau de la couche réseau, sans changer la taille du paquet IPv6, ce qui a pour conséquences :
  1. une transparence de bout-en-bout en évitant l'encapsulation (comme en MPLS) ou la fragmentation (comme en ATM) ;
  2. aucun conflit avec IPSec ;
  3. aucune interférence avec les protocoles de la couche de niveau 2 ;
  4. le champ `Hop Limit` est utilisé pour détecter les boucles sans recopier sa valeur dans un autre en-tête, comme en MPLS ;
  5. aucune violation du MTU[14].

Modèles d'acquisition d'étiquettes

L'architecture 6LSA définit trois modèles d'acquisition d'étiquettes mais un seul peut être utilisé à la fois dans un réseau 6LSA :

1. la création d'étiquettes locales qui sous-entend l'inspection des en-têtes IPv6 et de transport doit aussi s'assurer que le champ `Flow Label` est nul tant à l'entrée qu'à la sortie du réseau 6LSA ;

[14]Le *Maximum Transfer Unit* (MTU) est la taille maximale qu'un paquet peut avoir sur un chemin pour éviter toute fragmentation.

2. l'utilisation d'un protocole de distribution d'étiquettes tels que RSVP-TE ou le *Label Distribution Protocol* (LDP) qui est défini par Andersson *et al.* (2001) ;
3. la réutilisation d'étiquettes disponibles dans les en-têtes de paquets, ce qui n'est pas recommandé pour de grands réseaux.

Les routeurs de l'architecture 6LSA

Les routeurs faisant partie d'un réseau 6LSA comportent des fonctionnalités très semblables aux routeurs MPLS. En effet, les remarques de la section 2.3.1 pourraient très bien s'appliquer à 6LSA sauf en ce qui a trait aux points spécifiques à MPLS.

Différences entre les architectures 6LSA et MPLS

Nous avons mentionné au début de cette section que l'architecture 6LSA reprend l'idée maîtresse de MPLS en associant des étiquettes aux FEC. Par contre, il existe quelques différences importantes entre ces deux architectures :

- la propagation des étiquettes en MPLS se fait du nœud destination vers le nœud source, ce qui est l'inverse de 6LSA ;
- MPLS peut encapsuler des LSP grâce à une pile d'étiquettes alors que ce concept est inexistant dans 6LSA ;
- MPLS encapsule le paquet IPv6 original et ajoute un en-tête de 32 bits tandis que 6LSA ne modifie que le champ `Flow Label` de ce dernier ;
- 6LSA permet l'assignation opportuniste des étiquettes dont le concept ressemble à la création de flots par opportunisme décrit par Deering et Hinden (1995, sect. 6).

L'agrégation de flots

Un 6LSR peut fusionner des flots dans le but de réduire le nombre de flots à gérer. Par contre, l'absence du concept de pile d'étiquettes rend inutile la fusion de flots car pour différencier ceux-ci, un 6LSR devra consulter les adresses source et destination.

La sélection des routes

La méthode qui consiste à choisir le 6LSP pour une FEC en particulier est appelée "sélection de la route" et peut s'effectuer de deux façons :

1. chaque 6LSR choisit le prochain saut de façon indépendante en se basant sur les FEC disponibles[15] ;
2. pour le routage explicite, 6LSA permet l'utilisation d'un *Routing Header Extension* de IPv6 qui est considéré comme un attribut qu'une FEC en particulier doit inclure.

Résumé de 6LSA

L'architecture 6LSA permet de tirer profit des chemins commutés par étiquettes dans un réseau IPv6 pour la gestion de QdS et l'ingénierie de trafic. Nous avons comparé 6LSA avec MPLS en insistant sur leurs similitudes quoique certaines différences importantes les distinguent. Enfin, nous avons discuté des modèles d'acquisition d'étiquettes et de la sélection des routes.

2.6.2 L'architecture IPngLS

L'architecture IPngLS a été proposée par Roesler *et al.* (2002). Elle consiste à offrir les mêmes services que MPLS en stockant les données contenues dans l'en-tête MPLS (Figure 2.5) à même les champs de l'en-tête IPv6 (Figure 2.1) correspondants. La correspondance des champs MPLS dans les champs IPv6 est démontrée dans le Tableau 2.4.

TABLEAU 2.4 Correspondance des champs MPLS vers IPv6 dans IPngLS

Champ MPLS	Champ IPv6
`Label`	`Flow Label`
Exp	DSCP
TTL	`Hop Limit`
S	Hop-by-hop extension header

Détails sur la correspondance des champs

En ce qui concerne les champs `Label` et `TTL`, la correspondance est directe et n'implique aucune adaptation, puisque les champs IPv6 correspondants ont la même longueur et procurent exactement les mêmes fonctionnalités que les champs MPLS.

[15] Chakravorty mentionne à la section 7.13.1 que le *Hop-by-Hop Header Extension* peut être utilisé pour le routage mais sans donner de détails.

Par contre, le champ `Exp` est actuellement réservé pour usage expérimental dans le but d'offrir huit niveaux de priorités pour faciliter la gestion de QdS. Il serait facile d'identifier huit points de contrôle dans le champ `DSCP` de IPv6 qui comprend six bits.

La correspondance la plus difficile à effectuer est celle du bit `S` qui sert à identifier le bas de la pile d'étiquettes MPLS. En effet, il n'existe aucun champ IPv6 ayant la même fonction et aucun bit n'est libre dans l'en-tête IPv6. Pour régler ce problème, Roesler *et al.* proposent d'utiliser les *Hop-by-hop extension headers* dont l'utilisation pour stocker une étiquette est montrée à la Figure 2.6.

Le nombre d'octets requis pour accomoder un seul tunnel est 8, ce qui correspond au même nombre d'octets que pour MPLS. Ceci est vrai pour tous les nombres impairs de tunnels. Dans le cas de nombres de tunnels pairs, le nombre d'octets requis est supérieur de 4 octets par rapport à MPLS. Ce surplus est causé par le fait qu'une extension *Hop-by-Hop* a comme taille un multiple de 8 octets.

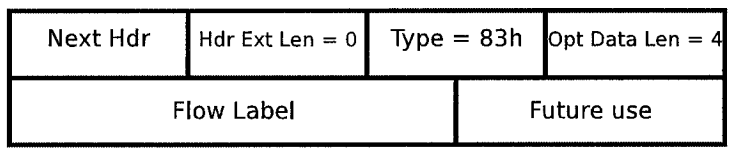

FIGURE 2.6 Hop-by-hop extension header pour un seul tunnel

Avantages de l'architecture IPngLS

Les avantages de l'architecture IPngLS sur l'architecture MPLS sont nombreux :

1. simplification du système par l'élimination d'un en-tête ;
2. diminution de la taille du paquet de 4 octets quand aucun tunnel n'est utilisé ;
3. simplification de la gestion de QdS par l'élimination du champ `Exp` de MPLS ;
4. simplication de la gestion du champ `TTL` en raison de la redondance du champ *Hop Limit* de IPv6.

Inconvénients de l'architecture IPngLS

Un certain nombre d'inconvénients sont rattachés à l'architecture IPngLS :

1. IPngLS ne peut être appliquée qu'à un réseau IPv6 ;
2. le champ `Flow Label` doit avoir la valeur 0 au nœud *ingress* du réseau ;

3. IPngLS requiert la lecture des 64 premiers bits de l'en-tête IPv6 contre 32 pour MPLS : ceci peut avoir un impact négatif pour les appareils portables ;
4. le concept de pile d'étiquette est supporté mais son utilisation est peu pratique.

Résumé de IPngLS

IPngLS permet de bénéficier des avantages de MPLS tout en simplifiant de façon notable l'architecture globale du système. La simplicité de cette approche ainsi que ses similitudes avec MPLS la rendent attrayante pour l'implémentation de chemins commutés par étiquette dans les réseaux IPv6.

2.6.3 Propagation de l'identificateur de flot

Les architectures 6LSA et IPngLS violent la règle établie par Deering et Hinden (1998) et Rajahalme *et al.* (2004) qui stipulent que la valeur du champ `Flow Label` ne doit pas être modifiée en cours de route. En effet, la valeur du `Flow Label` doit être préservée jusqu'à la destination finale du paquet.

Pour pallier à ce problème, il faut propager la valeur originelle du `Flow Label` jusqu'à la sortie du réseau. L'utilisation d'un protocole de distribution d'étiquettes pourrait propager cette valeur jusqu'au routeur *egress* qui serait responsable de restaurer la valeur originelle du `Flow Label` avant que le paquet ne poursuive son chemin.

2.7 Revue des problèmes à considérer

Nous avons vu que l'architecture *IntServ* gère la QdS sur des flots individuels avec précision. En contrepartie, elle force chaque routeur à conserver l'état des flots qui transitent par eux. Cette gestion additionnelle rend très difficile l'utilisation de cette architecture dans les WAN[16].

Dans un autre ordre d'idée, l'architecture *Diffserv* offre une façon simple, au niveau de chaque routeur, pour gérer la QdS de flots importants tels que des agrégats. En effet, les routeurs n'ont pas à gérer les états des flots individuels qui transitent par eux. Par contre, cette architecture est incapable de gérer des flots individuels à l'intérieur d'une même classe de service.

[16]Wide Area Networks.

Nous avons ensuite présenté l'architecture MPLS qui peut gérer aisément des agrégats par l'utilisation de chemins commutés par étiquette. On tire alors profit des circuits virtuels, aussi bien pour les réservations de ressources que pour la maximisation de l'utilisation des ressources du réseau. Enfin, la pile d'étiquettes permet de regrouper plusieurs classes de service en une même classe de service, créant ainsi des tunnels entre les routeurs.

Enfin, nous devons aussi considérer les difficultés supplémentaires engendrées par la mobilité des usagers. En effet, il faut déplacer les réservations de ressources d'un point d'accès à un autre. De plus, lorsque l'usager se connecte au réseau à l'aide d'un appareil sans-fil, les messages de rafraîchissement des états de réservation consomment une portion non-négligeable des ressources radio.

CHAPITRE 3

SOLUTION PROPOSÉE

Le Chapitre 1 a permis de fixer les objectifs de recherche qui devront être satisfaits dans le présent chapitre. Dans le Chapitre 2, nous avons passé en revue les différentes architectures permettant de gérer la QdS dans les réseaux IPv6. Nous avons aussi discuté des concepts d'ingénierie de trafic.

La solution recherchée devra permettre, dans le cadre du réseau d'accès d'un fournisseur de services pour usagers mobiles, de déplacer des réservations de ressources, des LSP, ainsi que d'autres paramètres appropriés à la technologie d'accès. De plus, cette solution devra minimiser les impacts de la relève intra-fournisseur sur la QdS rendue à l'usager mobile.

Nous commencerons donc par proposer une nouvelle architecture de transport permettant de créer des LSP par l'utilisation du champ `Flow Label` de IPv6. Nous aurons ainsi la capacité d'appliquer des concepts d'ingénierie de trafic afin de maximiser l'utilisation des ressources du réseau d'accès.

Lorsque le choix de l'architecture de transport sera complété, nous aborderons le problème du déplacement des réservations de ressources, des LSP et autres paramètres spécifiques à la technologie d'accès.

3.1 Proposition d'architecture de transport

Au terme de cette section, nous aurons une proposition d'architecture de transport permettant de créer des LSP en utilisant le champ `Flow Label` de IPv6.

Tout d'abord, nous passerons en revue les caractéristiques recherchées pour la nouvelle architecture de transport. Ensuite, nous procéderons à un choix et expliquerons les motivations qui le justifient. Enfin, nous explorerons les lacunes de ce choix et apporterons les solutions nécessaires.

3.1.1 Caractéristiques recherchées

L'architecture de transport recherchée doit posséder un ensemble de caractéristiques qui lui confèrent un avantage décisif sur les autres. Les caractéristiques recherchées sont détaillées dans les sous-sections qui suivent.

L'architecture doit être basée sur la commutation d'étiquettes

Les circuits virtuels sont adaptés aux besoins des applications temps-réel. En effet, la réception des paquets se fait dans l'ordre que les paquets ont été émis et la variation de délai de bout-en-bout est plus faible qu'en commutation de paquets.

De plus, une panne d'un lien ou d'un nœud du réseau n'entraîne pas inévitablement la perte de la communication, comme cela pourrait être le cas en commutation de circuit. En conséquence, la survivabilité du réseau, en cas de panne, est plus grande.

Compatibilité avec les mécanismes d'ingénierie de trafic

De par la nature compétitive de l'Internet, les fournisseurs de services sont forcés d'optimiser l'utilisation des ressources de leurs équipements.

Par ailleurs, les requêtes de réservations de ressources des applications temps-réel imposeront des contraintes strictes sur le réseau. Leur présence justifie l'importance de ce requis.

Simplicité et efficacité

L'architecture recherchée se veut simple à implémenter dans les routeurs. En effet, un routeur de grande capacité implémente un maximum de fonctionnalités en matériel (processeurs de réseau, FPGA) et ne prend que les décisions complexes en logiciel. Ainsi, afin de maximiser les performances, l'architecture doit être simple et efficace.

Il n'y a pas que le temps de traitement qui compte. En effet, on cherche, par la simplicité de l'approche, à réduire les impacts sur la bande passante.

Signification locale des étiquettes

L'architecture recherchée devrait favoriser l'évolutivité en ne donnant qu'une signification locale aux valeurs des étiquettes. De cette manière, on pourrait réutiliser les mêmes valeurs sur des liens ou des nœuds différents.

Classes de service

Une architecture de transport basée sur la commutation d'étiquettes doit associer une classe de service à chaque paquet. Une classe de service regroupe deux types d'informations :

1. la direction à prendre pour atteindre le nœud destination ;
2. les paramètres de QdS du paquet.

De cette façon, les informations liées à l'acheminement et à la classe de service sont comprises dans la valeur numérique de l'étiquette. On évite donc de répartir ces informations dans plusieurs champs du paquet.

Capacité de créer des LSP

La capacité de créer des LSP est primordiale dans les réseaux d'accès modernes en raison de la tendance croissante à développer des applications qui imposent des contraintes strictes sur le réseau.

Ces contraintes peuvent impliquer de choisir un chemin qui respecte les demandes en QdS des usagers en contournant un nœud congestionné, par exemple. Une autre motivation à la capacité de créer des LSP est le désir d'optimiser l'utilisation des ressources du réseau par le fournisseur de services.

Distribution des étiquettes

Les architectures qui peuvent se qualifier comme candidates potentielles pour notre environnement d'opération ne doivent pas limiter l'implantation d'un protocole de distribution d'étiquettes.

3.1.2 Évaluation des architectures retenues

Les architectures présentées dans la revue de littératures qui se qualifient pour une étude approfondie sont : MPLS[1], 6LSA et IPngLS.

[1] Nous considérerons MPLS même si cette architecture n'utilise pas le champ `Flow Label` pour stocker l'étiquette.

Évaluation de MPLS

L'architecture MPLS est bien connue et supportée par de nombreux fabricants d'équipements. Son principal avantage est de supporter n'importe quel protocole de la couche 3. En effet, cette information est elle-même encodée dans la classe de service puisqu'il n'y a pas de champ, dans l'en-tête MPLS, pour stocker un identificateur de protocole. La taille de son en-tête est de 32 bits.

Il est possible de construire une pile d'étiquettes en encapsulant un paquet MPLS à l'intérieur d'un autre. Cette particularité permet de construire des tunnels.

Évaluation de 6LSA

L'architecture 6LSA supporte les chemins commutés par étiquette en tirant profit du champ `Flow Label` de IPv6. La valeur du `Flow Label` doit être nulle au nœud *ingress* et est remise à zéro au nœud *egress.*

Le principal avantage de 6LSA est que la taille du paquet IPv6 n'est pas changée. De plus, puisque 6LSA n'intervient qu'au niveau de la couche 3, l'indépendance des couches de protocoles est favorisée.

L'architecture 6LSA ne permet pas de construire des tunnels en utilisant un concept de pile d'étiquettes. Par ailleurs, en cas d'agrégation de flots, l'opération inverse devient difficile à effectuer à cause de l'absence du concept de pile d'étiquettes.

Enfin, le sens de la propagation des étiquettes est l'inverse de celui de MPLS.

Évaluation de IPngLS

L'architecture IPngLS offre les mêmes services que MPLS. Cette architecture ne modifie en rien le fonctionnement de MPLS car les champs de son en-tête sont relocalisés à même les champs IPv6 correspondants. Il s'agit par contre d'une optimisation en bande passante lorsqu'il n'y a pas de pile d'étiquettes en utilisation dans le réseau, ce que l'auteur croit peu commun dans les réseaux d'accès.

Le concept de pile d'étiquettes en IPngLS est supporté par l'insertion d'un *Hop-by-hop extension header*, ce qui est plus complexe à traiter qu'une pile d'étiquettes en MPLS. Par contre, puisque l'utilisation d'une pile d'étiquettes n'est pas commune, son impact négatif est compensé par la réduction de 4 octets pour chaque trame, en l'absence d'une pile d'étiquettes.

Le principal avantage de IPngLS est la compatibilité directe avec tout ce qui est déjà utilisé avec MPLS. Une adaptation minimale peut toutefois être nécessaire.

Il faut noter que IPngLS, tout comme 6LSA, requiert que la valeur du `Flow Label` soit nulle au nœud *ingress* et, en conséquence, la remet à zéro au nœud *egress*.

Enfin, IPngLS ne peut être appliquée qu'à l'intérieur d'un réseau IPv6. Toutefois, puisque IPv6 est appelé à remplacer IPv4, ce désavantage d'aujourd'hui s'éliminera de lui-même dans le futur.

3.1.3 Comparaison des architectures retenues

Après avoir énoncé les avantages et désavantages de chaque architecture, nous devons maintenant proposer l'architecture de transport qui fera partie intégrante de la solution proposée dans ce mémoire.

Nous verrons qu'il est difficile de départager MPLS et IPngLS en raison de leurs ressemblances et parce que chacune possède un avantage non négligeable sur l'autre.

L'architecture 6LSA ne possède pas d'avantage décisif sur les deux autres candidates. De plus, elle ne supporte pas les piles d'étiquettes et requiert que le sens de propagation des étiquettes soit de la source vers la destination, ce qui est l'inverse de MPLS, et par extension, à IPngLS.

Cette architecture est encore à l'étape de recherche et ne constitue donc pas un choix sûr. En conséquence, elle ne sera donc pas retenue dans le cadre de ce mémoire.

L'architecture MPLS augmente la taille de chaque trame de 4 octets mais supporte de façon élégante le concept de pile d'étiquettes.

En contrepartie, les champs `TTL`, `Label` et `Exp` deviennent superflus lorsque le protocole de la couche 3 est IPv6.

L'architecture IPngLS a pour principal avantage de simplifier la pile de protocoles en enlevant l'en-tête MPLS. De plus, la taille de chaque trame est réduite de 4 octets en l'absence d'une pile d'étiquettes.

Par contre, le support de la pile d'étiquettes est fort complexe et devrait être évité si possible.

3.1.4 Choix de l'architecture et justifications

Suite à l'évaluation de la valeur propre de chacune des architectures ainsi qu'aux comparaisons entre elles, lorsque l'on considère :

1. les avantages et inconvénients de MPLS et IPngLS ;
2. le fait que le réseau d'accès n'utilise que IPv6 ;
3. le fait qu'un réseau d'accès ne soit pas un endroit où l'usage de tunnels est courant[2]...

Le choix d'architecture de transport pour le réseau d'accès s'arrête donc sur IPngLS. Toutefois, il nous faudra solutionner le problème de propagation de la valeur du `Flow Label`.

3.1.5 Le problème de propagation du Flow Label

L'architecture IPngLS implique que la valeur du champ `Flow Label` soit nulle au nœud *ingress* et qu'elle est remise à zéro au nœud *egress*. Dans IPngLS, aucun mécanisme n'existe pour propager le `Flow Label` du nœud *ingress* au nœud *egress*.

La propagation du `Flow Label` est essentielle pour assurer la transparence du mécanisme de commutation d'étiquettes et pour respecter l'esprit de la définition de ce champ (Deering et Hinden, 1998; Rajahalme *et al.*, 2004).

Applications tirant profit d'un Flow Label

Il est important de rappeler que seules les applications qui ont des contraintes de débit et de délai de bout-en-bout devraient en général réserver des ressources.

Les protocoles RTP et RTCP (Schulzrinne *et al.*, 2003) sont conçus pour les applications temps-réel. Il sont définis au niveau application et ainsi n'utilisent que des numéros de ports réservés à celles-ci[3].

Les numéros de ports des sessions temps-réel d'une application sont généralement assignés dynamiquement par le système d'exploitation. En conséquence, ces applications doivent recourir aux services d'un protocole de signalisation.

[2]Les tunnels sont surtout utilisés pour l'agrégation de flots et la création de VPN.

[3]Les numéros de ports système sont compris dans l'intervalle 0–1023. En contrepartie, les applications utilisent des numéros de ports compris entre 1024 et 65535.

Utilisation d'un protocole de signalisation

L'utilisation d'un protocole de signalisation permet d'initier autant de sessions indépendantes que requises pour établir une communication multimédia. En effet, une session est caractérisée par le type de média, ses besoins en ressources, ports, etc.

Le protocole de signalisation proposé par l'IETF est SIP (Rosenberg *et al.*, 2002). Ce dernier fait appel au protocole SDP (Handley et Jacobson, 1998) pour décrire chaque session dont les paramètres sont énoncés en format texte.

L'avènement de IPv6 a mené à une mise à jour de SDP (Olson *et al.*, 2002). Toutefois, aucun paramètre n'a été déclaré pour identifier le `Flow Label` puisque l'utilisation de ce dernier est toujours considérée expérimentale.

En contrepartie, le protocole SDP peut être étendu pour ajouter un nouvel attribut à la description de l'émetteur. En conséquence, un nouvel attribut indiquant la valeur du champ `Flow Label` de la source peut être ajouté.

Propagation du Flow Label dans le réseau d'accès

L'implantation de LSP dans le réseau d'accès encourage l'utilisation d'un protocole de distribution d'étiquettes. Par ailleurs, le protocole recherché devra permettre de réserver des ressources.

En conséquence, si la description du flot comprend un `Flow Label`, nous pourrons propager cette valeur vers le BES par l'ajout d'un objet destiné à cet effet. Dans le cas contraire, nous assumerons que la valeur est zéro. Une autre approche serait de modifier un objet existant afin d'y incorporer la valeur initiale du `Flow Label`.

Nous reviendrons plus tard sur ce point précis, lorsque nous aborderons la structure du protocole de réservation de ressources.

3.1.6 Conclusions sur l'architecture de transport

Nous avons choisi l'architecture IPngLS car elle offre les mêmes fonctionnalités de base que MPLS tout en étant plus simple. Nous avons aussi évoqué deux solutions pour la propagation de la valeur initiale du `Flow Label`, au travers du réseau d'accès, à l'aide d'un protocole de distribution d'étiquettes. La première solution consiste à ajouter un objet qui contient la valeur initiale du `Flow Label` tandis que la seconde approche consiste à modifier un objet existant.

3.2 Modèle pour les réservations de ressources et la distribution d'étiquettes

Cette section vise à proposer un modèle de réservation de ressources et de distribution d'étiquettes qui fera partie intégrante de la proposition de solution finale.

Le modèle proposé prend la forme d'un protocole orienté récepteur fortement inspiré de RSVP-TE ainsi que de HPMRSVP (Abondo, 2005) pour les réservations de ressources et la distribution d'étiquettes.

3.2.1 Justification du choix de HPMRSVP

Le protocole possède un ensemble de caractéristiques qui justifient son choix dans le cadre de ce mémoire :

- il adresse spécifiquement les besoins en déplacements de réservations de ressources lors de la relève ;
- il limite les réservations de ressources au réseau d'accès ;
- il assigne au AES le rôle de préserver les états des réservations effectuées par l'appareil mobile ;
- il est basé sur l'architecture HMIPv6.

3.2.2 Justification du choix de RSVP-TE

Le choix de RSVP-TE comme base de la solution finale repose sur les points suivants :

- RSVP-TE est compatible avec l'architecture MPLS[4] ;
- RSVP-TE est techniquement équivalent à CR-LDP (Jamoussi *et al.*, 2002) mais le premier est plus populaire auprès des équipementiers[5] ;
- une décision de l'IETF (Anderson et Swallow, 2003) a statué que le support de deux protocoles équivalents représente une duplication de travail et a tranché en faveur de RSVP-TE, au détriment de CR-LDP qui ne sera plus développé ;
- HPMRSVP utilise une syntaxe dérivée de RSVP, ce qui facilitera l'intégration de ses fonctionnalités avec RSVP-TE.

[4]Cette contrainte doit être satisfaite puisque nous avons choisi IPngLS dans le réseau d'accès.
[5]Une comparaison détaillée entre CR-LDP et RSVP-TE est faite par Wang (2001, sect. 4.5.4).

3.2.3 Conclusion sur la forme du modèle

Dans cette section, nous avons déterminé la forme que prendra le nouveau protocole qui serait déployé dans un réseau d'accès pour appareils mobiles.

Ce protocole combine la capacité de réservation de ressources à un mécanisme de distribution d'étiquettes. En conséquence, puisque le protocole proposé dérive directement de HPMRSVP et RSVP-TE, le nom qui lui sera attribué est HPMRSVP-TE.

3.3 Le protocole HPMRSVP-TE

Cette section présente un nouveau protocole de réservation de ressources, basé sur la commutation d'étiquettes, développé pour un réseau d'accès pour usagers mobiles.

Tout d'abord, nous introduirons quelques considérations d'ordre général. Ensuite, nous décrirons en détail chacun des objets qui composent ces messages. Enfin, nous présenterons les formats des messages HPMRSVP-TE.

Dans les section suivantes, nous employerons la notation couramment utilisée dans les documents RFC de l'IETF (voir Bradner, 1997). Lorsque nécessaire, les termes décrits dans ce document seront indiqués, en caractères gras et entre parenthèses, à la suite d'une spécification afin de mettre l'accent sur l'importance de celle-ci.

3.3.1 Traitement des messages par chaque routeur

Les messages HPMRSVP-TE doivent (**MUST**) être traités par chaque routeur afin de procéder à la réservation de ressources. Ainsi, chaque entité IPv6 en jeu doit (**MUST**) supporter le "Router Alert Option" (Partridge et Jackson, 1999). De plus, un nouvel identificateur de contenu, distinct de celui qui est utilisé pour RSVP, serait souhaitable pour spécifier le protocole HPMRSVP-TE.

Par ailleurs, l'IANA doit réserver un identificateur de protocole pour le champ `Next Header` de IPv6 qui soit différent de celui utilisé pour RSVP.

3.3.2 Utilisation du multicasting

Le protocole HPMRSVP-TE n'est pas conçu pour opérer sur des groupes de nœuds ; seules les sessions *unicast* sont supportées. Le support pour le multicasting est laissé en travaux futurs.

3.3.3 Format des messages HPMRSVP-TE

Le format des messages HPMRSVP-TE débute par un en-tête commun suivi d'un ensemble d'objets distincts ayant chacun un en-tête. Les formats d'en-tête commun et d'en-tête d'objet sont identiques à ceux définis pour RSVP.

Format de l'en-tête commun

La description de l'en-tête commun est détaillée ci-dessous et réfère à la Figure 2.3 :

`Ver :` `4 bits`
Version du protocole. La version actuelle est 1.

`Flags :` `4 bits`
Aucun Flag n'est défini. L'émetteur doit (**MUST**) mettre le champ à zéro et le récepteur doit (**MUST**) ignorer ce champ.

`Msg Type :` `8 bits`
Type de message (voir le Tableau 3.1).

`Checksum :` `16 bits`
Complément à un du complément à un de la somme des champs de l'en-tête, avec le champ Checksum à 0 pour des fins de calcul. La valeur 0 signifie qu'aucun Checksum n'a été transmis. Si le résultat du calcul du Checksum donne 0, la valeur 0xFFFF doit être stockée dans ce champ.

`TTL :` `8 bits`
Valeur originelle du champ `TTL` utilisée pour transmettre ce message.

`Reserved :` `8 bits`
Réservé pour usage futur. L'émetteur doit (**MUST**) mettre le champ à zéro et le récepteur doit (**MUST**) ignorer ce champ.

`Length :` `16 bits`
Longueur totale du message en octets, incluant l'en-tête commun et tous les objets de longueur variable.

TABLEAU 3.1 Types de messages définis dans HPMRSVP-TE

Numéro	Message	Numéro	Message	Numéro	Message
1	Path	4	ResvErr	7	ResvConf
2	Resv	5	PathTear	8	PathMod
3	PathErr	6	ResvTear	10	PathModErr

Format de l'en-tête d'objet

La description de l'en-tête d'objet est détaillée ci-dessous et réfère à la Figure 2.4 :

`Length :` `16 bits`
Longueur totale de l'objet en octets, incluant l'en-tête. La longueur doit (**MUST**) être un multiple de 4 et avoir une valeur d'au moins 4.

`Class-Num :` `8 bits`
Identificateur de la classe d'objet (voir le Tableau 3.2).

`C-Type :` `8 bits`
Type de l'objet, unique à l'intérieur d'une classe.

Ordre d'apparition des objets d'un message

À moins d'indication contraire, le format d'un message est présenté selon une suite d'objets dont l'ordre d'apparition devrait (**SHOULD**) être respecté. En contrepartie,

TABLEAU 3.2 Classes d'objets définies dans HPMRSVP-TE

Numéro	Classe	Numéro	Classe
0	NULL	11	SENDER_TEMPLATE
1	SESSION	12	SENDER_TSPEC
3	RSVP_HOP	13	ADSPEC
4	INTEGRITY	14	POLICY_DATA
5	TIME_VALUES	16	LABEL
6	ERROR_SPEC	19	LABEL_REQUEST
8	STYLE	20	EXPLICIT_ROUTE
9	FLOWSPEC	21	RECORD_ROUTE
10	FILTER_SPEC	207	SESSION_ATTRIBUTE

une implémentation de HPMRSVP-TE doit (**MUST**) pouvoir interpréter les objets dans n'importe quel ordre.

3.3.4 Description des objets du protocole HPMRSVP-TE

Chaque message du protocole HPMRSVP-TE se compose d'objets indépendants. On retrouve certains des objets décrits ci-dessous dans plusieurs messages différents. Nous présenterons les formats des messages HPMRSVP-TE suite à la description des objets qui les composent. Pour tous les objets présentés, chaque ligne d'en-tête représente 32 bits de données.

L'objet `NULL` porte le numéro de classe 0 et son champ `C-Type` doit (**MUST**) être ignoré. Il peut apparaître n'importe où dans la séquence des objets et son contenu doit (**MUST**) être ignoré par le récepteur.

L'objet `SESSION` permet d'identifier une session en particulier et est présenté à la Figure 3.1. Il est inclus dans tous les messages et spécifie l'adresse destination du tunnel. Il contient aussi deux identificateurs de tunnel qui demeurent constants pour la durée de vie du tunnel. Le champ `Source Home Address` contient le HoA du nœud source tandis que le champ `Destination Home Address` contient le HoA du nœud destination. Le champ `Flow Label` contient la valeur initiale du `Flow Label` IPv6 qui doit être restaurée à la sortie du tunnel.

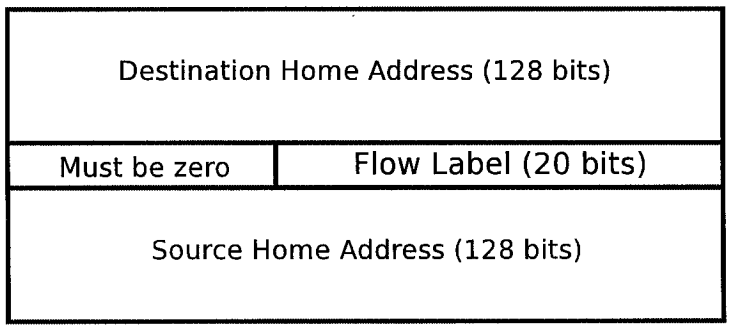

FIGURE 3.1 L'objet `SESSION` (`Class` = 1, `C_Type` = 9)

L'objet `RSVP_HOP` contient l'adresse IPv6 du nœud qui a envoyé le message dans lequel il se trouve. On y trouve aussi un identificateur d'interface dont la valeur n'est utile qu'au nœud qui a envoyé le message ; cet identificateur permet au nœud précédent (*Previous Hop, PHOP*) de retracer l'interface par laquelle le message à été envoyé. Le format de l'objet est présenté à la Figure 3.2.

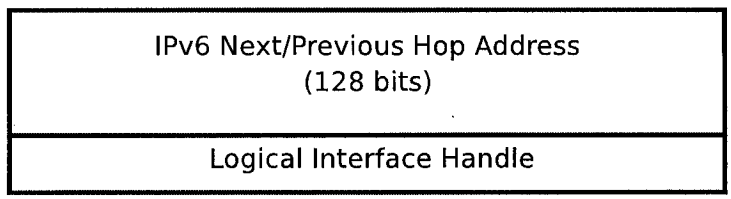

FIGURE 3.2 L'objet `RSVP_HOP` (`Class` = 3, `C_Type` = 2)

L'objet `INTEGRITY` est une signature cryptographique qui garantit l'intégrité du message et l'authentification de l'émetteur. Si cet objet est présent dans un message, il doit (**MUST**) être situé immédiatement à la suite de l'en-tête commun. La description complète de l'objet `INTEGRITY` est faite par Baker *et al.* (2000)[6].

L'utilisation de l'objet `INTEGRITY` dépasse le cadre de recherche de ce mémoire. En conséquence, nous mentionnons son existence en raison de son importance mais nous n'en tiendrons pas compte dans nos travaux.

L'objet `TIME_VALUES` spécifie la période de rafraîchissement, en millisecondes, utilisée par le créateur du message. La Figure 3.3 montre le format de cet objet.

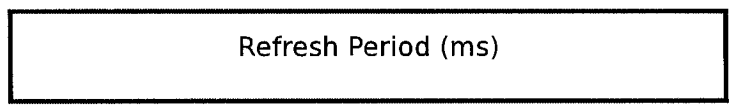

FIGURE 3.3 L'objet `TIME_VALUES` (`Class` = 5, `C_Type` = 1)

[6]En raison d'un conflit de numéro d'identification de message avec les optimisations de rafraîchissement des sessions RSVP (Berger *et al.*, 2001), les numéros associés aux messages `Challenge` et `Integrity Response` ont été modifiés par Braden et Zhang (2001).

L'objet `ERROR_SPEC` indique qu'une erreur s'est produite et contient des informations qui permettent au récepteur d'identifier la cause de l'erreur. La Figure 3.4 montre le format de cet objet. Le champ `Flags` est réservé à un usage futur : il doit (**MUST**) être mis à 0 par l'émetteur et être ignoré à la réception. Les codes d'erreurs sont définis dans la spécification de RSVP (Braden *et al.*, 1997, Annexe B).

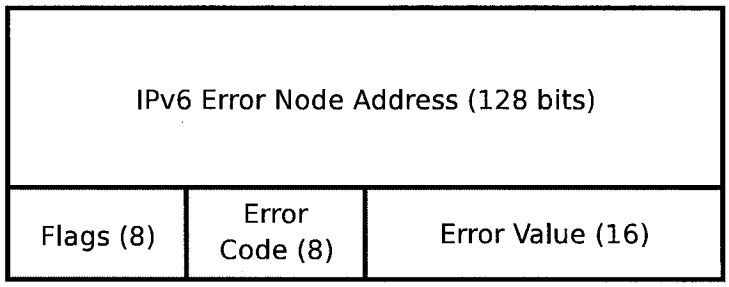

FIGURE 3.4 L'objet `ERROR_SPEC` (`Class` = 6, `C_Type` = 2)

L'objet `STYLE` définit le style de réservation ainsi que les informations spécifiques au style qui ne sont pas comprises dans les objets `FLOWSPEC` et `FILTER_SPEC`. La Figure 3.5 montre le format de cet objet. Le champ `Option Vector` est décrit dans le Tableau 3.3. On notera que seules les réservations explicites sont supportées.

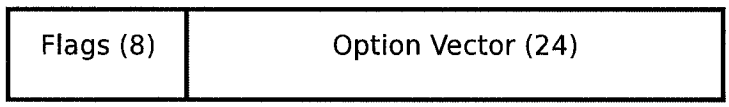

FIGURE 3.5 L'objet `STYLE` (`Class` = 8, `C_Type` = 1)

TABLEAU 3.3 Le champ `Option Vector` de l'objet `STYLE`

Champ	Bits	Description
Reserved	19	Réservé pour usage futur
Sharing Control	2	Réservations distinctes (01) ou partagées (10)
Sender Selection Control	3	Émetteurs explicites (010)

L'objet `FLOWSPEC` définit les paramètres de QdS qui ont été réservés. Le format de l'objet dépend du modèle de service demandé. La Figure 3.6 montre l'objet `FLOWSPEC` lorsque le modèle *Guaranteed Service* a été choisi. En contrepartie, la Figure 3.7 montre l'objet `FLOWSPEC` lorsque le modèle *Controlled Load Service* est utilisé.

Dans le Tableau 3.4, on retrouve une brève description des champs de l'objet `FLOWSPEC`, qui ont été marqués d'une lettre, pour la Figure 3.6 et la Figure 3.7.

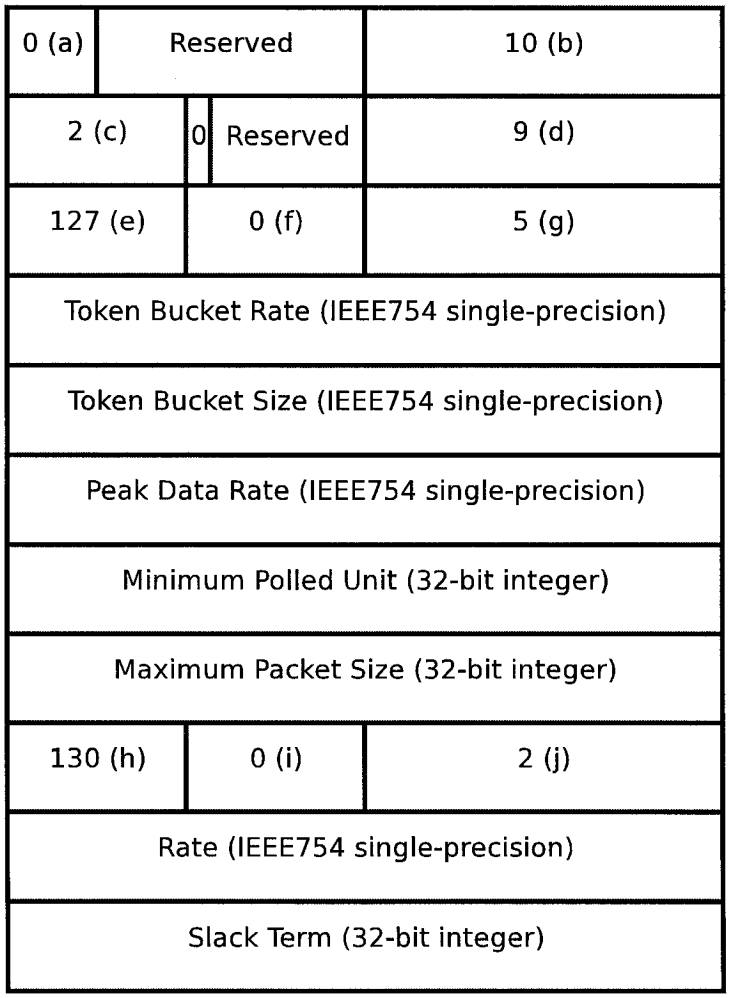

FIGURE 3.6 L'objet `FLOWSPEC` (*Guaranteed Service*, `Class` = 9, `C_Type` = 2)

TABLEAU 3.4 Champs de l'objet FLOWSPEC (Figures 3.6 et 3.7)

Lettre	Description
a	Version du format de message (=0)
b	Longueur totale en mots de 32 bits sans l'en-tête
c	En-tête de service, représente le numéro de service
d	Longueur de la structure du service en mots de 32 bits sans l'en-tête
e,h	Identificateur de paramètre (voir Shenker et Wroclawski, 1997a)
f,i	Flags du paramètre courant
g,j	Longueur de la structure du paramètre en mots de 32 bits sans l'en-tête

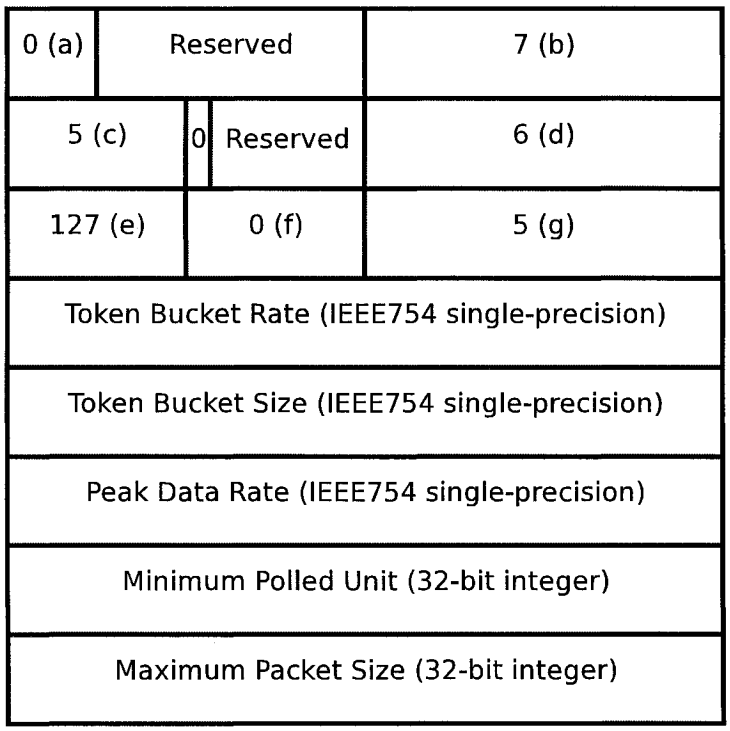

FIGURE 3.7 L'objet FLOWSPEC (*Controlled-Load Service*, Class = 9, C_Type = 2)

L'objet `FILTER_SPEC` indique quels sont les paquets qui recevront la QdS désirée. Il comprend l'adresse source de l'émetteur ainsi qu'un identificateur unique de tunnel. La Figure 3.8 montre le format de cet objet.

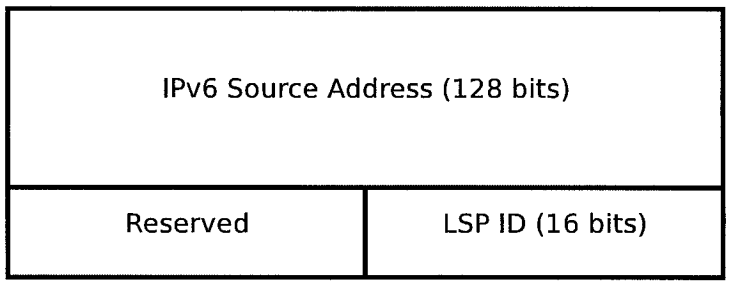

FIGURE 3.8 L'objet `FILTER_SPEC` (`Class` = 10, `C_Type` = 8)

L'objet `SENDER_TEMPLATE` décrit les paquets émis par la source. Son format est identique à celui de l'objet `FILTER_SPEC` (Figure 3.8) sauf que son numéro de classe est distinct (`Class` = 11, `C_Type` = 8).

L'objet `SENDER_TSPEC` spécifie les caractéristiques du trafic émis par la source. Le format de cet objet est identique à celui de l'objet `FLOWSPEC` utilisé pour le *Controlled-Load Service* (Figure 3.7), à l'exception du numéro de service qui a la valeur 1.

L'objet `ADSPEC` comporte les informations relatives au mécanisme OPWA. Chaque nœud modifie cet objet pour y intégrer ses propres caractéristiques. La Figure 3.9 montre le format de cet objet composé de fragments spécifiques à chaque service. Le Tableau 3.5 décrit brièvement le contenu des champs de l'objet `ADSPEC`. La description des fragments est élaborée par Wroclawski (1997a, sect. 3.3).

TABLEAU 3.5 Champs de l'objet `ADSPEC`

Lettre	Description
a	Version du format de message (=0)
b	Longueur totale en mots de 32 bits sans l'en-tête
c,d,e	Fragments de données (voir Wroclawski, 1997a)

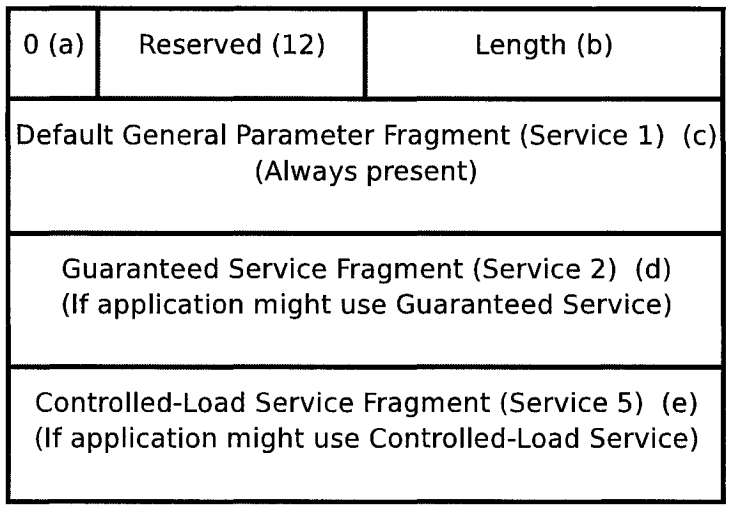

FIGURE 3.9 L'objet ADSPEC (Class = 13, C_Type = 2)

L'objet POLICY_DATA est utilisé pour transporter des informations relatives aux politiques d'admission d'une requête de réservation de ressources. La Figure 3.10 montre le format de cet objet ; son usage est décrit par Herzog (2000).

L'une des informations pouvant être transportée dans un objet POLICY_DATA est l'élément de priorité de préemption d'un flot (Herzog, 2001) dont le format est détaillé à la Figure 3.11. Cet élément indique l'importance relative du flot courant lorsqu'il y a compétition pour l'obtention des ressources. La réservation d'un flot de moindre importance peut donc être annulée pour faire place à un flot plus important.

On peut aussi insérer dans un objet POLICY_DATA des informations permettant d'identifier de façon sécuritaire l'utilisateur et/ou l'application qui effectue la réservation de ressources. Cet élément est décrit par Yadav *et al.* (2001) et son format est montré à la Figure 3.12. Le champ P_Type prend la valeur 2 pour identifier un usager tandis que la valeur 3 est utilisée pour l'identification d'une application. La description d'attributs servant à décrire une application est faite par Bernet et Pabbati (2000).

Enfin, on peut insérer dans un objet POLICY_DATA des attributs servant à la validation d'une session individuelle. Cet élément est décrit par Hamer *et al.* (2003) et son format est montré à la Figure 3.13.

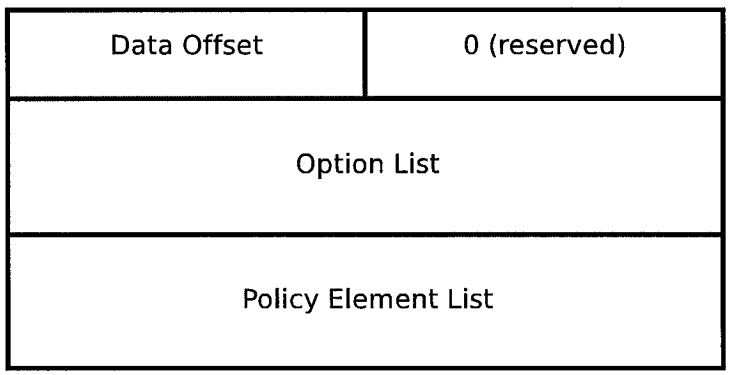

FIGURE 3.10 L'objet POLICY_DATA (Class = 14, C_Type = 1)

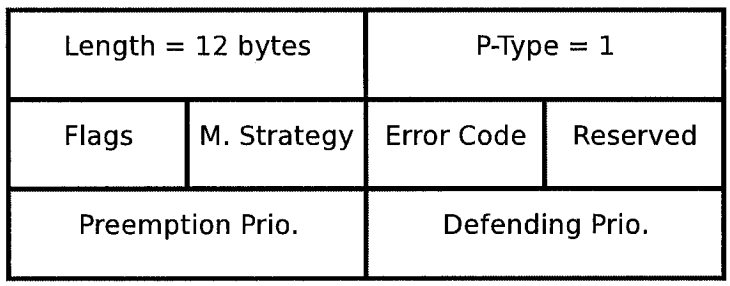

FIGURE 3.11 L'élément de politique PREEMPTION_PRI (P_Type = 1)

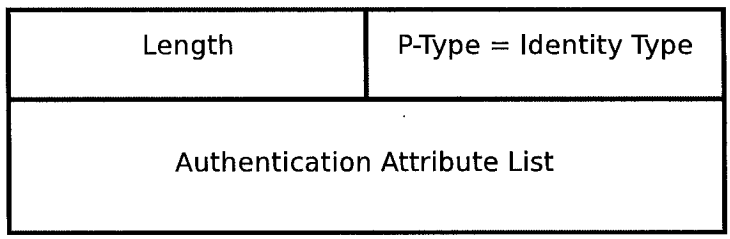

FIGURE 3.12 L'élément de politique AUTH_DATA (P_Type = 2/3)

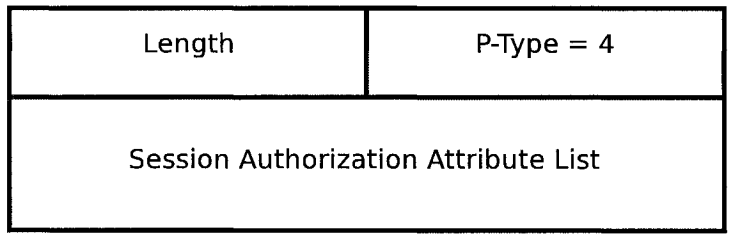

FIGURE 3.13 L'élément de politique AUTH_SESSION (P_Type = 4)

L'objet RESV_CONFIRM peut être inclus dans un message *Resv* ou *ResvConf* afin de transporter l'adresse IP d'un nœud qui désire obtenir une confirmation de traitement. La Figure 3.14 montre le format de cet objet.

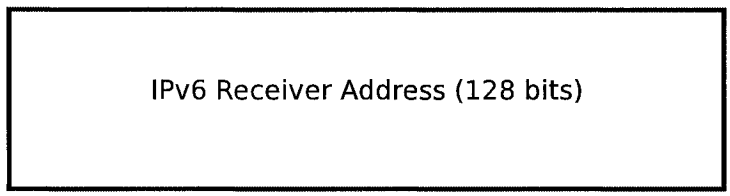

FIGURE 3.14 L'objet RESV_CONFIRM (Class = 15, C_Type = 2)

L'objet LABEL contient l'étiquette allouée par le nœud en aval. On retrouve cet objet dans le message *Resv*. Le format de cet objet est présenté à la Figure 3.15.

Pour les styles de réservation FF et SE, un objet LABEL est associé à chaque émetteur. Chaque LABEL doit (**MUST**) immédiatement suivre l'objet FILTER_SPEC qui lui correspond.

FIGURE 3.15 L'objet LABEL (Class = 16, C_Type = 1)

L'objet `LABEL_REQUEST` constitue une requête d'assignation d'étiquette envoyée en direction du nœud *egress*. Le format de cet objet est présenté à la Figure 3.16.

Le champ `L3PID` spécifie le numéro de protocole qui circule au niveau 3, selon les valeurs `EtherType`[7]. Puisque nous n'utilisons que IPv6, la valeur de ce champ doit (**MUST**) être `0x86DD`. Le champ `Reserved` doit (**MUST**) avoir la valeur zéro.

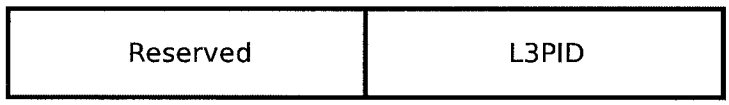

FIGURE 3.16 L'objet `LABEL_REQUEST` (`Class` = 19, `C_Type` = 1)

L'objet `EXPLICIT_ROUTE` permet au nœud *ingress* de spécifier les adresses des routeurs à traverser afin de choisir le trajet qu'empruntera le flot de données. Le format de cet objet est montré à la Figure 3.17 tandis que le format des sous-objets est présenté à la Figure 3.18. Chacun de ces sous-objets contient l'adresse IPv6 du prochain routeur sur le trajet. Un attribut `Loose` indique si la présence du prochain routeur est requise (stricte) ou non (souple).

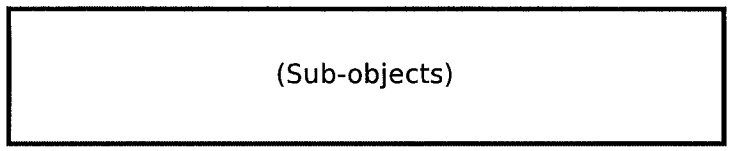

FIGURE 3.17 L'objet `EXPLICIT_ROUTE` (`Class` = 20, `C_Type` = 1)

L'objet `RECORD_ROUTE` peut être utilisé par l'émetteur pour obtenir le trajet emprunté par le flot de données ou pour découvrir dynamiquement tout changement de trajet. Le format de cet objet est identique à celui de l'objet `EXPLICIT_ROUTE` (Figure 3.17) sauf que le numéro de classe est distinct (`Class` = 21, `C_Type` = 1). Cet objet est composé de sous-objets dont le format est détaillé à la Figure 3.19.

[7]Ces valeurs sont disponibles dans une base de données de l'IANA (voir Reynolds, 2002).

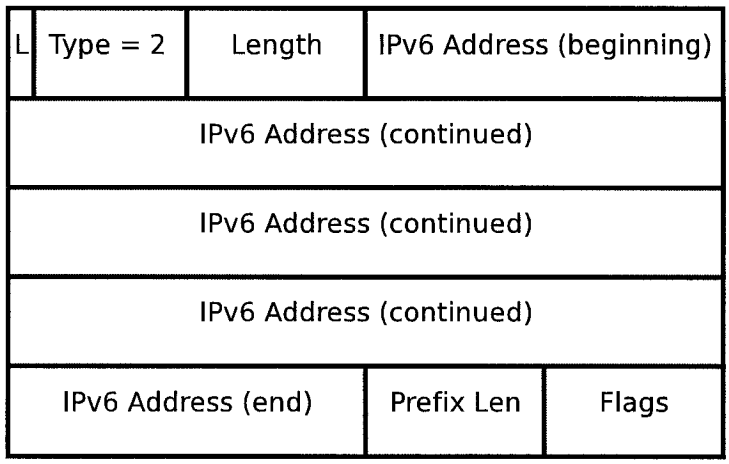

FIGURE 3.18 Le sous-objet `IPv6 Prefix` de l'objet `EXPLICIT_ROUTE`

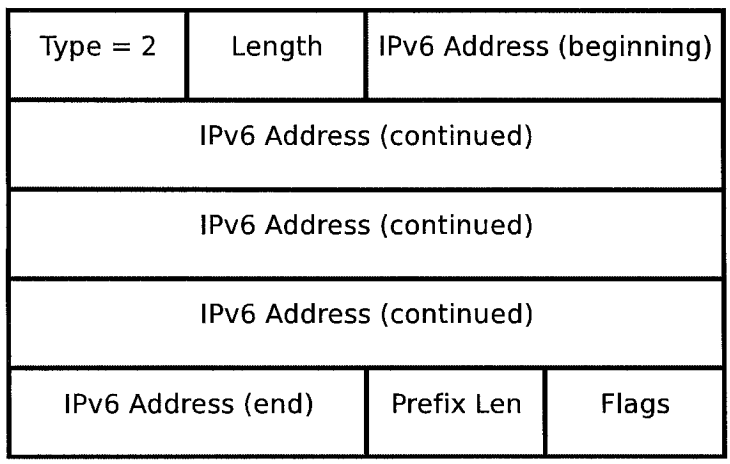

FIGURE 3.19 Le sous-objet `IPv6 Address` de l'objet `RECORD_ROUTE`

L'objet `SESSION_ATTRIBUTE` peut être ajouté au message *Path* afin de faciliter l'identification de la session. En plus du nom associé à la session, des informations de contrôle supplémentaires sont incluses :

- la priorité de la session à réserver des ressources en cas de congestion au niveau d'un nœud en aval ;
- la priorité de la session à conserver une réservation de resource existante en cas de congestion d'un nœud ;
- les affinités envers les classes de ressources (Awduche *et al.*, 1999, sect. 5.6.3) qui peuvent définir des contraintes de routage supplémentaires ;
- la demande de protection locale en cas de défaillance.

La Figure 3.20 montre le format de l'objet lorsque les affinités de ressources ne sont pas supportées. En contrepartie, la Figure 3.21 montre le format de l'objet lorsque les affinités de ressources sont spécifiées. Enfin, le Tableau 3.6 décrit les champs de l'objet `SESSION_ATTRIBUTE`.

Le support des priorités d'établissement et de préservation de session est optionnel (**OPTIONAL**). Un nœud ne devrait pas (**SHOULD NOT**) modifier le contenu de ces champs à la réception. De plus, la valeur du champ `Setup Priority` ne devrait pas (**SHOULD NOT**) être supérieure à la valeur du champ `Holding Priority`.

Le support de la protection locale est optionnel (**OPTIONAL**) au niveau des nœuds. Il serait toutefois souhaitable d'ajouter à HPMRSVP-TE les extensions définies par Pan *et al.* (2005) si ces dernières n'entraînent pas d'incompatibilité. En effet, le nœud source du LSP protégé est déplacé lors de la relève. Par contre, ces extensions permettraient de rerouter automatiquement un LSP ou un groupe de LSP en cas de panne de nœud ou de lien puisque les réservations de ressources seraient garanties.

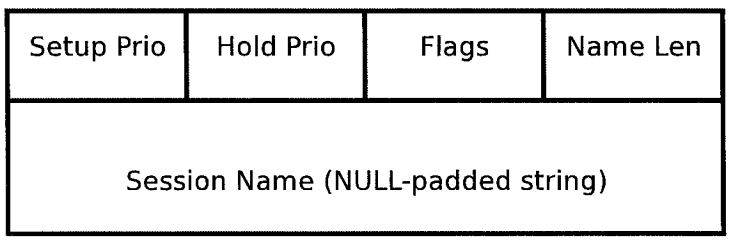

FIGURE 3.20 L'objet `SESSION_ATTRIBUTE` sans affinités (`Class` = 207, `C_Type` = 7)

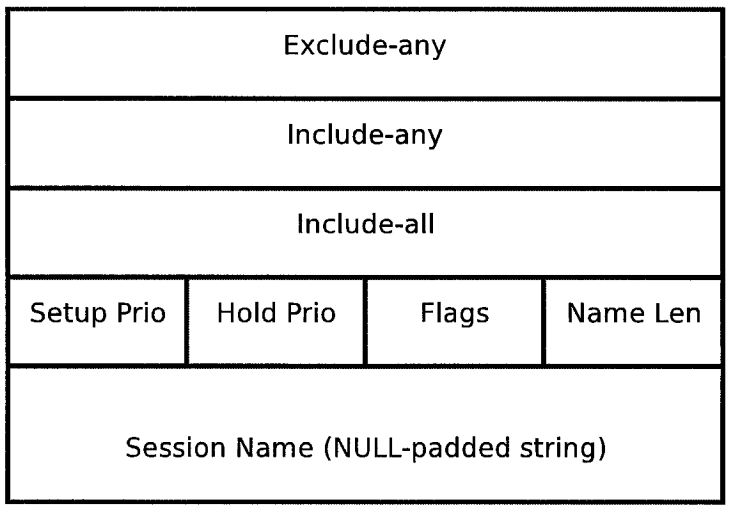

FIGURE 3.21 L'objet SESSION_ATTRIBUTE avec affinités (Class = 207, C_Type = 1)

TABLEAU 3.6 Champs de l'objet SESSION_ATTRIBUTE (Figures 3.20 et 3.21)

Champ	Description
Setup Priority	Priorité de la session à réserver des ressources (valeur de 0 à 7, 0 étant la plus haute priorité)
Holding Priority	Priorité de la session à conserver une réservation (valeur de 0 à 7, 0 étant la plus haute priorité)
Flags	voir Tableau 3.7
Name Length	Longueur du nom de session en octets
Session Name	Chaîne de caractères complétée avec NULL
Exclude-any	Filtre d'attributs de 32 bits qui entraînent le rejet du lien
Include-any	Filtre d'attributs de 32 bits qui acceptent le lien
Include-all	Filtre d'attributs de 32 bits qui acceptent le lien

TABLEAU 3.7 Attributs du champ Flags de l'objet SESSION_ATTRIBUTE

Bit	Attribut	Description
0	Local protection desired	Lors d'une panne, permet à un nœud de trouver un chemin alternatif
1	Label recording desired	Permet l'enregistrement des étiquettes, en plus du chemin, lors d'un RECORD_ROUTE
2	SE Style desired	Indique que le nœud *ingress* peut rerouter le tunnel sans le détruire

3.3.5 Sémantique des messages du protocole HPMRSVP-TE

Le protocole HPMRSVP-TE se compose d'un ensemble de messages dont les définitions seront données ci-dessous. Le format de chaque message est présenté sous la forme de *Backus-Naur*, augmentée de crochets pour indiquer les objets optionnels (Crocker et Overell, 1997).

Le message *Path* instaure l'état d'un nouveau chemin ou modifie l'état d'un chemin existant. Un état de chemin comprend au minimum la description de la session, l'adresse du nœud précédent, la période de rafraîchissement, une requête d'étiquette ainsi que les paramètres du trafic émis par la source. Il constitue le premier message d'un échange en HPMRSVP-TE.

```
<Path Message> ::= <Common Header> [<INTEGRITY>]
                   <SESSION> <RSVP_HOP> <TIME_VALUES>
                   [<EXPLICIT_ROUTE>] <LABEL_REQUEST>
                   [<SESSION_ATTRIBUTE>]
                   [<POLICY_DATA>...]
                   <SENDER_TEMPLATE> <SENDER_TSPEC>
                   [<ADSPEC>] [<RECORD_ROUTE>]
```

Le message *PathErr* signifie qu'il y a eu une erreur lors du traitement du message *Path*. Ce message ne modifie en rien l'état de réservation ou de rafraîchissement en cours. Il est envoyé à l'émetteur du message *Path* en suivant le chemin inverse.

```
<PathErr Message> ::= <Common Header> [<INTEGRITY>]
                      <SESSION> <ERROR_SPEC>
                      [<POLICY_DATA>...]
                      <SENDER_TEMPLATE> <SENDER_TSPEC>
```

Le message *Resv* est émis par le destinataire afin de réserver des ressources. Il suit le chemin inverse du message *Path* qui lui est associé. Il faut noter que les objets `LABEL` et `RECORD_ROUTE` (si présents), sont reliés à l'objet `FILTER_SPEC` qui les précède. Au plus un objet de chaque type (`LABEL` ou `RECORD_ROUTE`) peut suivre chaque objet `FILTER_SPEC`.

```
<Resv Message> ::= <Common Header> [<INTEGRITY>]
                   <SESSION> <RSVP_HOP> <TIME_VALUES>
                   [<RESV_CONFIRM>] [<POLICY_DATA>...]
                   <STYLE> <flow desc list>

<flow desc list> ::= <FF flow desc list> | <SE flow desc>

<FF flow desc list> ::= <FLOWSPEC> <FILTER_SPEC>
                        <LABEL> [<RECORD_ROUTE>] |
                        <FF flow desc list> <FF flow desc>

<FF flow desc> ::= [<FLOWSPEC>] <FILTER_SPEC>
                   <LABEL> [<RECORD_ROUTE>]

<SE flow desc> ::= <FLOWSPEC> <SE filter spec list>

<SE filter spec list> ::= <SE filter spec> |
                          <SE filter spec list>
                          <SE filter spec>

<SE filter spec> ::= <FILTER_SPEC> <LABEL>
                     [<RECORD_ROUTE>]
```

Le message *ResvErr* signifie qu'il y a eu une erreur lors du traitement du message *Resv* ou qu'une réservation est annulée spontanément, par exemple pour des raisons administratives. Les règles entourant l'utilisation du message *ResvErr* sont identiques à celles de RSVP (Braden *et al.*, 1997, sect. 3.1.8).

```
<ResvErr Message> ::= <Common Header> [<INTEGRITY>]
                      <SESSION> <RSVP_HOP> <ERROR_SPEC>
                      [<POLICY_DATA>...]
                      <STYLE> [<error flow descriptor>]

<error flow descriptor> ::= <FF flow descriptor> |
                            <SE flow descriptor>
```

Le message *PathTear* élimine les états de chemin et de réservations associés à une session. La libération des ressources par ce message devrait (**SHOULD**) toujours être effectuée afin de rendre disponibles les ressources pour d'autres usages.

```
<PathTear Message> ::=  <Common Header> [<INTEGRITY>]
                        <SESSION> <RSVP_HOP>
                        <SENDER_TEMPLATE> <SENDER_TSPEC>
```

Le message *ResvTear* élimine les états de réservations associés à une session. La libération des ressources par ce message devrait (**SHOULD**) toujours être effectuée afin de rendre disponibles les ressources pour d'autres usages.

```
<ResvTear Message> ::=  <Common Header> [<INTEGRITY>]
                        <SESSION> <RSVP_HOP> <STYLE>
                        <flow descriptor list>
```

Le message *PathMod* ne peut être utilisé que lorsqu'une session n'est pas prise en charge par un protocole de signalisation. En effet, l'utilisation d'un protocole de signalisation permet de charger dans le BES les paramètres de QdS du lien descendant, après vérification auprès du serveur AAA[8].

Dans le cas où aucun protocole de signalisation n'est utilisé, le message *PathMod* est envoyé au BES, par le MN, dans le but d'initier une réservation de ressources sur le lien descendant au nom du CN. Ce message peut aussi servir à modifier une réservation de ressources existante sur le lien descendant au nom du CN.

Le message *PathMod* n'est transmis au BES, par les routeurs intermédiaires, qu'afin de traiter l'objet `INTEGRITY`. Il peut être transmis directement au BES si cet objet n'est pas supporté. Dans ce cas, on retire le "Router Alert Option" de l'en-tête IPv6 du message.

La définition du message *PathMod*, pour le protocole HPMRSVP-TE, est une version considérablement différente de celle décrite par Abondo (2005). En effet, la version originale permettait au MN de modifier les paramètres de QdS aussi bien sur les liens montants que descendants. De plus, ce message était transmis de bout-en-bout. Toutefois, cette utilisation du message *PathMod* complique inutilement l'implémentation du protocole dans les routeurs et le BES.

[8] Authentication, Authorization and Accounting.

Dans un autre ordre d'idée, la décision de limiter le message *PathMod* au réseau d'accès où le lien descendant doit être modifié permet de limiter les impacts d'une négociation de bout-en-bout. Ainsi, les modifications de réservations requises peuvent s'effectuer en parallèle, dans les deux réseaux d'accès.

Par ailleurs, le MN et le CN sont tous deux impliqués dans la modification de réservation puisqu'ils doivent renégocier les paramètres de QdS à l'aide d'un protocole de signalisation ou à même le protocole qui contrôle la session dans le cas contraire.

Enfin, pour des raisons de sécurité, seul le nœud mobile qui se trouve dans un réseau d'accès donné peut demander de modifier une réservation de ressource, que ce soit sur le lien montant ou descendant.

Lorsque le BES reçoit un message *PathMod*, celui-ci doit vérifier si la modification de réservation est autorisée. Si oui, ce dernier émet un message *Path* en direction de la source afin d'allouer de nouvelles ressources. Ensuite, lorsque le BES reçoit le message *Resv* correspondant, celui-ci peut procéder à la libération des ressources inutilisées.

Dans le cas où la modification de réservation est refusée, le BES doit (**MUST**) envoyer un message *PathModErr* à la source de la requête.

```
<PathMod Message> ::=  <Common Header> [<INTEGRITY>]
                       <SESSION> <RSVP_HOP>
                       [<POLICY_DATA>...]
                       <SENDER_TSPEC>
```

Le message *PathModErr* signale une erreur survenue pendant le traitement du message *PathMod* par le BES. Le contenu de ce message est semblable à celui du message *PathErr* mais n'inclut pas d'objet `SENDER_TEMPLATE`.

Ce message n'est pas interprété par les routeurs intermédiaires hormis le traitement de l'objet `INTEGRITY`. Ainsi, si le support de cet objet est inexistant, le BES peut envoyer ce message directement à la source du message *PathMod* en retirant le "Router Alert Option" de l'en-tête IPv6 du message.

```
<PathModErr Message> ::=  <Common Header> [<INTEGRITY>]
                          <SESSION> <ERROR_SPEC>
                          [<POLICY_DATA>...]
                          <SENDER_TSPEC>
```

Le message *ResvConf* signale que le traitement du message *Resv* correspondant a été effectué avec succès. Le message est envoyé à l'adresse *unicast* du récepteur.

Ce message n'est pas interprété par les routeurs intermédiaires hormis le traitement de l'objet `INTEGRITY`. Ainsi, dans le cas où le support de cet objet est inexistant, le BES peut envoyer ce message directement à la source du message *Resv* en retirant le "Router Alert Option" de l'en-tête IPv6 du message.

```
<ResvConf Message> ::=  <Common Header> [<INTEGRITY>]
                        <SESSION> <ERROR_SPEC> <RESV_CONFIRM>
                        <STYLE> <flow descriptor list>
```

3.4 Réservations de ressources

Nous avons présenté les formats des objets HPMRSVP-TE pour ensuite détailler les formats des messages du protocole. Nous allons maintenant présenter les échanges de messages visant à effectuer les réservations initiales des ressources ainsi que les modifications de ces dernières.

3.4.1 La réservation initiale des ressources

L'utilisation d'un protocole de signalisation, tel que SIP, permet l'échange des paramètres de la session[9]. Dans le cas d'une session multimédia SIP, les paramètres sont échangés à l'aide du protocole SDP.

Une fois la signalisation de session complétée, les paramètres de la session sont transférés aux routeurs d'accès et BES concernés, et ce dans chacun des deux réseaux d'accès impliqués. Ainsi, les routeurs d'accès accepteront les requêtes de réservations de ressources pour les liens montants, de la part du MN et du CN. De plus, les BES respectifs pourront effectuer les réservations de ressources pour les liens descendants, sur réception de la requête de réservation pour le lien montant correspondant.

Les sessions qui ne sont pas prises en charge par un protocole de signalisation obligent le MN et le CN à demander à leur BES respectif de réserver des ressources sur le lien descendant, au nom de leur correspondant, grâce au message *PathMod*.

La Figure 3.22 montre la réservation initiale des ressources dans le réseau d'accès du MN. On remarque que le BES effectue la réservation de ressources au nom du CN sur le lien descendant.

[9]Les paramètres de la session comprennent la méthode d'encodage, le débit, les ports, etc.

Contrairement à Abondo (2005), dans le cas où le MN et le CN se trouvent dans le même réseau d'accès, nous forcerons les flots de données à passer par le BES pour des fins de sécurité et de contrôle. En effet, il existe une tendance dans l'industrie qui vise à centraliser les mécanismes de protection, de façon à protéger efficacement les usagers d'un réseau d'accès contre des attaques provenant des autres usagers de ce même réseau. En conséquence, une réservation initiale, qu'elle implique des nœuds mobiles dans le même réseau d'accès ou non, s'effectue de la même façon.

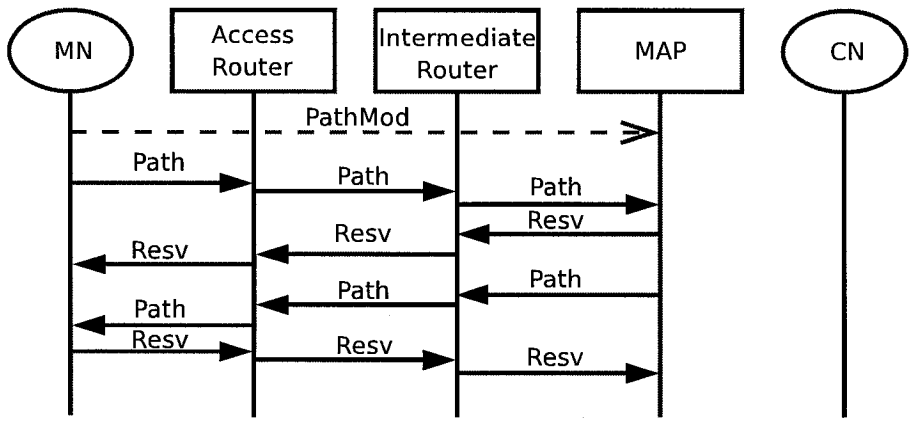

FIGURE 3.22 Réservation initiale des ressources pour le MN

3.4.2 Modification des paramètres de QdS d'une session

Au cours d'une session entre le MN et le CN, ceux-ci peuvent demander une modification des paramètres de QdS. La Figure 3.23 montre comment s'effectue la modification des paramètres de QdS d'une session allant du MN vers le CN.

On remarque que le message *PathMod* est limité au réseau d'accès du CN. Ainsi, ce message déclenche la nouvelle réservation de ressources au niveau du BES. Par ailleurs, ce message comprend les paramètres de QdS pour le lien descendant.

Il est important de mentionner que le mécanisme de modification des paramètres de QdS demeure identique si les deux nœuds impliqués se trouvent dans le même réseau d'accès.

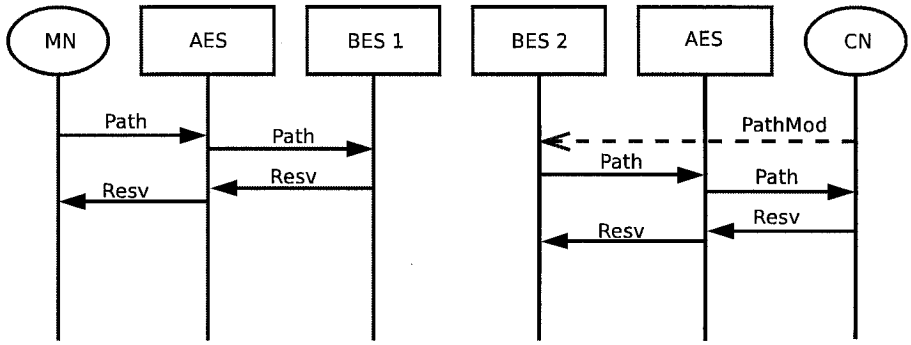

FIGURE 3.23 Modification d'une réservation de ressources

3.5 Les relèves intra-domaines

Nous verrons dans cette section comment se déroulent les divers scénarios de relèves intra-domaines de niveau 3. Il est toutefois important de formuler au préalable les hypothèses suivantes :

- tous les nœuds supportent les fonctionnalités de F-HMIPv6 et HPMRSVP-TE ;
- l'anticipation de la relève est supportée par des mécanismes déclencheurs de niveau 2 appropriés ;
- le BES possède l'information nécessaire à la relève dans le domaine HMIPv6, incluant l'adresse de niveau 2 (ou l'identificateur) et le préfixe réseau des AES ;
- les réservations initiales de ressources sont complétées entre le MN et le BES.

Les scénarios étudiés supposent que la relève peut être initiée par le MN ou par le réseau. De plus, nous considérerons la possibilité que le BES supporte le bicasting pendant toute la durée de la relève de niveau 3. Le bicasting est un mécanisme supporté par le BES qui permet, pendant toute la durée de la relève de niveau 3, d'envoyer chaque paquet destiné au MN aux adresses PCoA et NCoA de ce dernier.

3.5.1 La relève initiée par le mobile sans bicasting

La Figure 3.24 montre la procédure de relève lorsqu'elle est initiée par le mobile et sans support du bicasting de la part du BES. Ainsi, pendant la relève, les paquets destinés au MN seront accumulés par le NAES, conformément à la procédure F-HMIPv6. Notre procédure est différente de celle proposée par Abondo (2005) à la Figure 3.5 de sa thèse :

1. la première modification consiste à libérer les ressources sur le chemin allant du PCoA du MN vers le BES, via le PAES. Ainsi, lorsque le tunnel bidirectionnel entre le BES et le NAES est établi, les ressources maintenant inutilisées sont libérées pour d'autres usages, cela préalablement à la déconnexion du MN au niveau du PAES ;
2. la seconde modification consiste à retirer les réservations de ressources entre le BES et le NAES[10]. En effet, les réservations de ressources proposées par Abondo ne regénèrent pas les réservations entre le NAES et le MN. De plus, les réservations en question supposent l'utilisation d'un mécanisme de transfert de contexte entre les routeurs d'accès qui est absent dans le diagramme ;
3. lorsque le BES envoie le message *FBACK*, le seul message qui peut être envoyé pour rétablir les réservations de ressources du MN au NAES est un message *Path* sur le lien descendant. En effet, le BES possédait les paramètres de QdS du lien descendant avant le début de la relève. Ce message est traité par le NAES puis accumulé dans ses tampons en attendant que le MN se manifeste ;
4. lorsque le MN se connecte au NAES, celui-ci envoie le message *RS* (Router Sollicitation) avec l'option *FNA* afin de déclencher le transfert des paquets accumulés au niveau du NAES. Le message *Path* pour le lien descendant est alors envoyé au MN. Puisque la procédure F-HMIPv6 stipule que les paquets accumulés dans les tampons du NAES sont priorisés, il n'est pas nécessaire que les réservations de ressources soient effectives à ce moment ;
5. les réservations HPMRSVP-TE sont ensuite effectuées normalement avant de procéder à l'échange des messages *LBU* et *LBACK* qui détruisent le tunnel entre le BES et le NAES.

On peut supposer que la dégradation de QdS résultant de la relève pourrait être de courte durée, à moins que de nombreux paquets soient accumulés par le NAES pendant la relève. Par ailleurs, une portion de la bande passante pourrait être réservée par l'opérateur du réseau pour assurer une QdS adéquate, et ce pendant toute la durée de la relève.

[10]Une méthode envisageable serait de prioriser le trafic destiné au NAES pendant la relève.

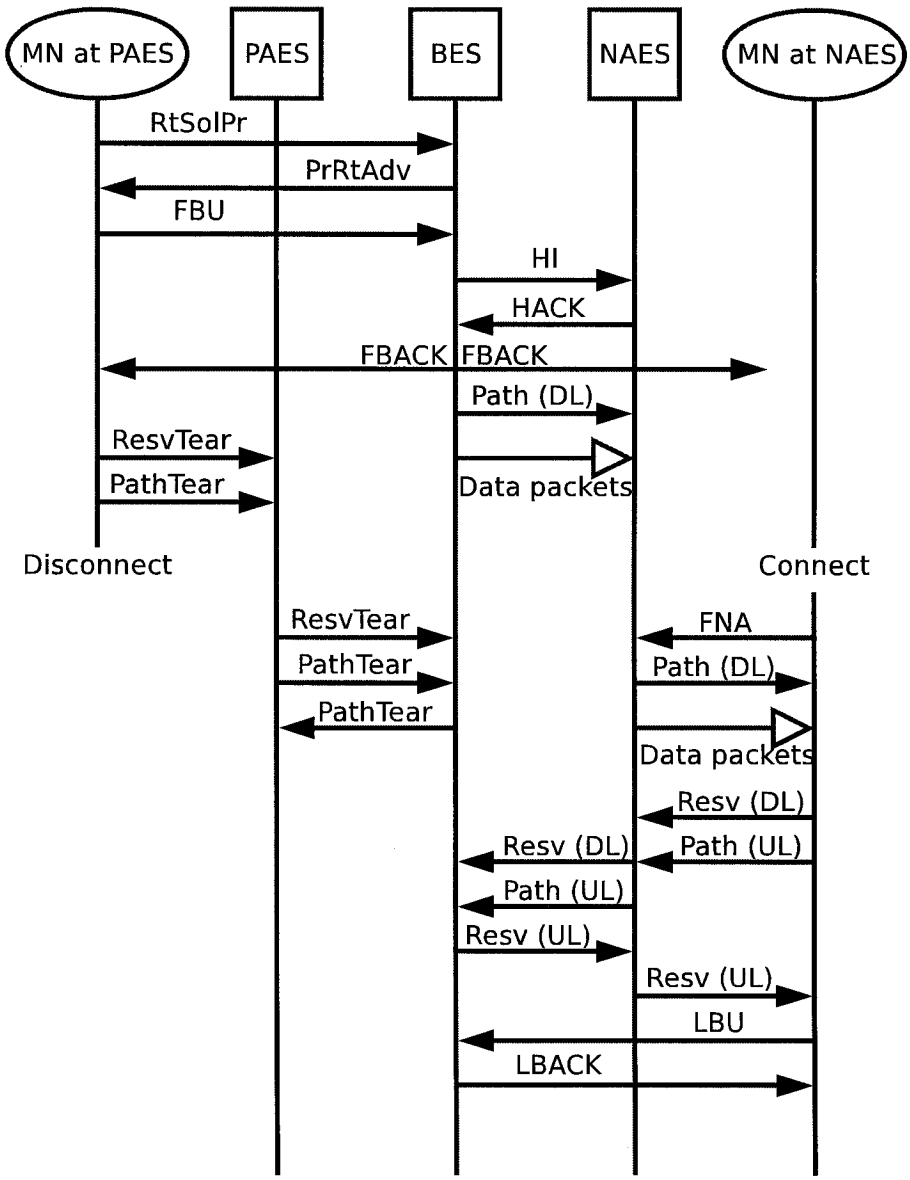

FIGURE 3.24 Relève initiée par le MN sans bicasting

3.5.2 La relève initiée par le mobile avec bicasting

La Figure 3.25 montre la procédure de relève lorsqu'elle est initiée par le mobile, avec support du bicasting de la part du BES.

La principale différence avec le scénario sans bicasting est que le message *FBU* initie le bicasting plutôt que la création d'un tunnel entre le BES et le NAES.

La relève avec bicasting soulève un problème de synchronisation car le BES doit s'assurer que le MN est connecté au NAES avant d'envoyer un message *Path* pour le lien descendant.

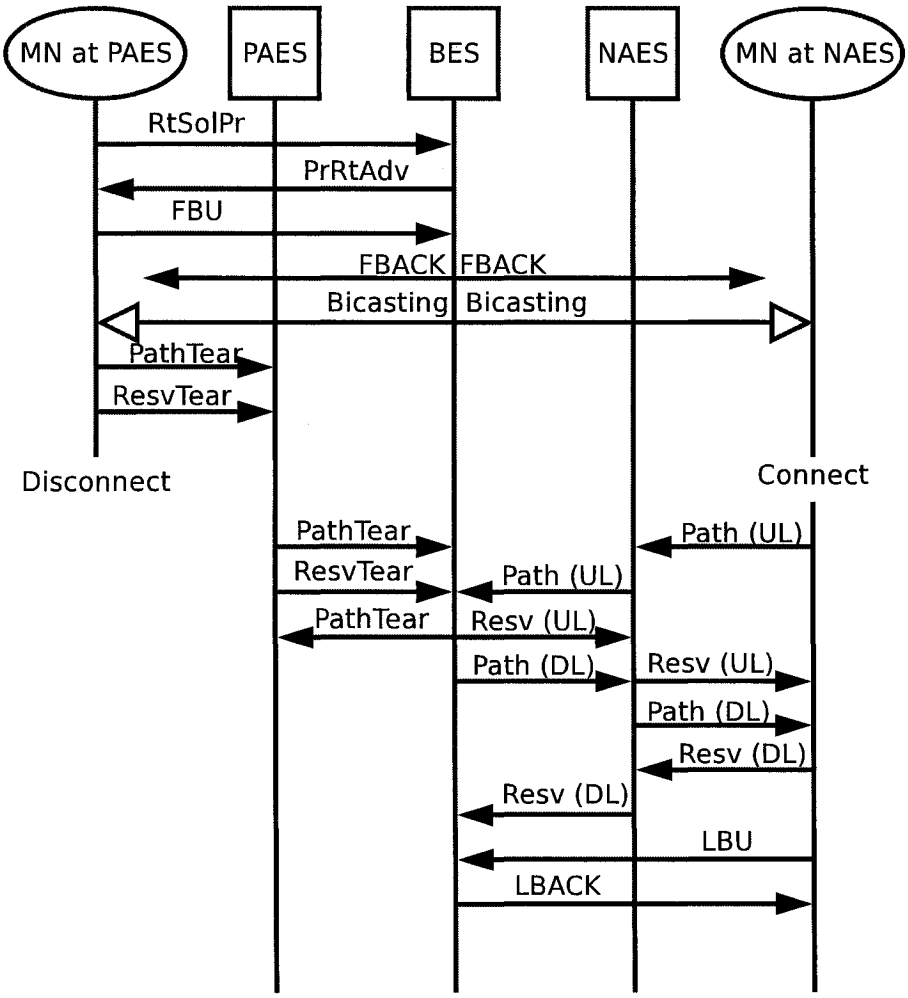

FIGURE 3.25 Relève initiée par le MN avec bicasting

3.5.3 La relève initiée par le réseau

La Figure 3.26 montre la procédure de relève lorsqu'elle est initiée par le réseau, avec ou sans support du bicasting de la part du BES.

On remarque que la relève est déclenchée par un mécanisme de niveau 2 par le PAES ou le NAES. Ce déclencheur peut provenir du PAES, pour indiquer une baisse critique du niveau de puissance du lien radio qui force la relève, ou du NAES pour signifier la réception d'un signal plus puissant de la part de ce dernier.

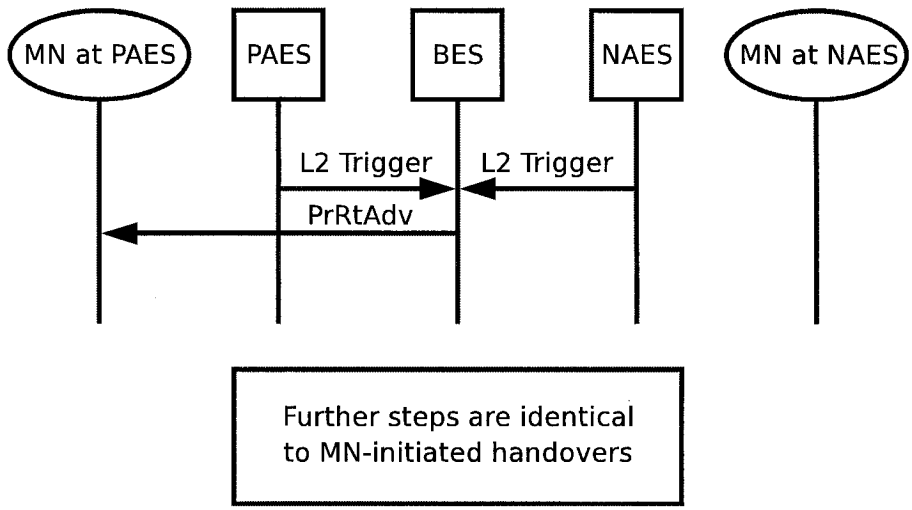

FIGURE 3.26 Relève initiée par le réseau

3.6 Mécanisme de rafraîchissement

Nous devons maintenant assurer l'évolutivité de notre proposition de solution en réduisant la charge de notre protocole sur le réseau. En effet, le mécanisme de rafraîchissement standard de RSVP impose une charge importante au réseau sans-fil dans lequel le MN se trouve. Cette charge de travail supplémentaire draine de précieuses ressources radio et réduit la durée de vie des piles qui alimentent le MN.

Abondo (2005) propose que le routeur d'accès maintienne les états de réservations au nom du MN. Ainsi, une plus grande portion des ressources radio demeurera disponible pour offrir des services aux abonnés. L'idée est que le routeur d'accès perçoit

la présence du MN par des mécanismes de niveau 2. Il peut ainsi réduire la charge de traitement de ce dernier en maintenant lui-même les états des réservations en cours.

L'implémentation de ce mécanisme suppose que les routeurs d'accès soient configurés de telle sorte qu'ils effectuent le rafraîchissement des états des réservations. De plus, pour le lien descendant, un routeur d'accès doit éviter de propager les messages de rafraîchissement vers le MN.

3.7 Conclusions sur la solution proposée

Dans ce chapitre, nous avons présenté les deux volets de notre solution. Le premier volet consistait à choisir une architecture de transport tandis que le second portait sur la conception d'un protocole de réservation de ressources et de distribution d'étiquettes adapté aux réseaux d'accès pour usagers mobiles.

L'architecture de transport recherchée devait présenter tous les avantages de l'architecture MPLS, tout en étant plus simple. En ce sens, nous avons proposé d'utiliser l'architecture IPngLS. Toutefois, l'architecture IPngLS comporte une faille importante en ce qui a trait à la propagation du champ `Flow Label` de IPv6 au travers du réseau d'accès. Nous avons identifié deux avenues de solutions possibles.

Nous avons ensuite porté notre attention sur la conception d'un protocole de réservation de ressources et de distribution d'étiquettes afin de permettre la gestion de la QdS par le biais de l'ingénierie de trafic. Ce protocole s'inspire de HPMRSVP et RSVP-TE pour offrir des fonctionnalités adaptées aux besoins présents et futurs des opérateurs de réseaux.

Nous avons vu que les principales caractéristiques de ce protocole sont de permettre le déplacement des réservations de ressources lors de la relève et la capacité d'intégrer les concepts d'ingénierie de trafic. De plus, les réservations de ressources se limitent au réseau d'accès plutôt que de s'étendre de bout-en-bout.

Le dernier point dont nous avons traité consiste en l'optimisation de l'utilisation des ressources radio. Nous avons mentionné l'implémentation du mécanisme de rafraîchissement proposé par Abondo (2005). Ce mécanisme assure l'évolutivité du protocole HPMRSVP-TE.

Enfin, il est important de noter que les objets de HPMRSVP-TE qui furent ajoutés ou modifiés, d'après les définitions de RSVP et RSVP-TE, doivent se conformer à la procédure de modification de RSVP élaborée par Kompella et Lang (2004).

CHAPITRE 4

ANALYSE DE PERFORMANCE

Dans le chapitre précédent, nous avons proposé un protocole de réservation de ressources et de distribution d'étiquettes pour un réseau d'accès avec usagers mobiles.

Dans un premier temps, nous bâtirons un modèle analytique de notre protocole afin d'évaluer les probabilités de blocage, d'interruption forcée et de compléter avec succès une session temps-réel.

Ensuite, en nous basant sur un réseau de test, nous procéderons à l'évaluation des délais et du nombre de messages échangés pour les scénarios suivants :

1. la réservation initiale des ressources ;
2. la modification d'une réservation de ressources existante ;
3. le rafraîchissement des états de chemin et de réservation ;
4. la relève intra-domaine sans bicasting.

4.1 Construction d'un modèle analytique

La première étape de notre processus de validation consiste à élaborer un modèle mathématique des caractéristiques de notre protocole. Ainsi, nous pourrons évaluer les probabilités de blocage, d'interruption forcée et compléter une session avec succès dans un environnement contrôlé.

Tout d'abord, nous émettrons les hypothèses sous-jacentes à notre analyse et présenterons le réseau d'accès qui servira d'environnement d'évaluation. Ensuite, nous définirons les paramètres du modèle analytique. Enfin, nous dévoilerons les expressions mathématiques des probabilités.

4.1.1 Constantes et variables du modèle analytique

Avant de dévoiler les expressions mathématiques qui constituent la base de notre modèle analytique, il convient d'établir un certain nombre de définitions. Les va-

riables et constantes qui seront utilisées dans les expressions mathématiques élaborées ultérieurement sont présentées dans le Tableau 4.1.

TABLEAU 4.1 Constantes et variables du modèle analytique

Symbole	Description
λ	Taux d'arrivée moyen des requêtes de réservation de ressources
$\frac{1}{\mu}$	Durée moyenne d'une session
C	Capacité d'une cellule (nombre de sessions supportées)
v_{moy}	Vitesse moyenne des MN dans le réseau d'accès
L	Longueur d'un côté d'une cellule carrée
n	Nombre moyen de MN dans une cellule
ρ	Charge d'une cellule
P_b	Probabilité de blocage d'une requête de réservation
P_f	Probabilité d'interruption forcée d'une session
P_c	Probabilité de compléter une session avec succès

4.1.2 Hypothèses et environnement d'évaluation

Pour des raisons de simplification d'analyse, notre modèle analytique demeurera valide tant que les hypothèses suivantes seront respectées :

- les réservations de ressources sont toutes de la même taille et la capacité d'une cellule est un multiple de cette taille ;
- la distribution des MN dans les cellules du réseau d'accès est uniforme ;
- le réseau a atteint un état de régime permanent ;
- les MN se déplacent à une vitesse moyenne v_{moy}, horizontalement ou verticalement seulement, sans combinaison linéaire de ces mouvements.

Comme réseau d'accès, nous utiliserons une grille carrée de 8×8 cellules telle que présentée à la Figure 4.1. Afin de simplifier le modèle, nous considèrerons que toutes les cellules sont techniquement identiques et de même taille. De plus, chaque cellule C_{xy} est reliée à un seul routeur d'accès AR_{xy}. Les routeurs d'accès sont tous reliés à un seul BES, sans aucun routeur intermédiaire.

La grille d'un réseau d'accès est repliée sur elle-même comme une tore. En effet, un déplacement vers le haut lorsqu'un MN se trouve dans la rangée 0 le fera réapparaître à la rangée 7. Le même principe est applicable aux autres arêtes de la grille.

C_{00}	C_{01}	C_{02}	C_{03}	C_{04}	C_{05}	C_{06}	C_{07}
C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}	C_{17}
C_{20}	C_{21}	C_{22}	C_{23}	C_{24}	C_{25}	C_{26}	C_{27}
C_{30}	C_{31}	C_{32}	C_{33}	C_{34}	C_{35}	C_{36}	C_{37}
C_{40}	C_{41}	C_{42}	C_{43}	C_{44}	C_{45}	C_{46}	C_{47}
C_{50}	C_{51}	C_{52}	C_{53}	C_{54}	C_{55}	C_{56}	C_{57}
C_{60}	C_{61}	C_{62}	C_{63}	C_{64}	C_{65}	C_{66}	C_{67}
C_{70}	C_{71}	C_{72}	C_{73}	C_{74}	C_{75}	C_{76}	C_{77}

FIGURE 4.1 Grille de 8×8 représentant un réseau d'accès

4.1.3 Paramètres du modèle analytique

Nous devons maintenant décrire les paramètres de notre modèle analytique :

- le *temps moyen d'inter-arrivée des requêtes* de réservation de ressources suit une loi exponentielle dont la moyenne est $\frac{1}{\lambda}$;
- le *temps de maintien* d'une réservation de ressources suit une loi exponentielle dont la moyenne est $\frac{1}{\mu}$;
- la *charge d'une cellule* est la fraction des ressources utilisées dans celle-ci, exprimée par l'équation 4.1 ;

$$\rho = \frac{n\lambda}{C\mu} \tag{4.1}$$

- la *probabilité de blocage* P_b indique la proportion des requêtes de réservations de ressources qui sont rejetées ;
- la *probabilité d'interruption forcée* P_f indique la proportion des sessions qui doivent être annulées suite à une relève, faute de ressources dans la nouvelle cellule ;
- la *probabilité de compléter une session* P_c indique la proportion des sessions qui se terminent avec succès, sans égard au nombre de relèves subies.

4.1.4 Calcul des probabilités

Dans cette section, nous élaborerons les expressions mathématiques des probabilités de blocage, d'interruption forcée et de l'achèvement avec succès d'une session.

Probabilité de blocage

Pour le calcul de cette probabilité, nous utiliserons le modèle de trafic *Erlang-B*. En conséquence, nous supposerons que le rejet d'une requête de réservation n'entraîne pas de nouvelle tentative.

De plus, nous avons précédemment émis l'hypothèse que toutes les réservations de ressources sont de la même taille et que la capacité d'une cellule est un multiple de cette taille. Ainsi, si l'on se réfère au système téléphonique traditionnel, on peut considérer chaque réservation de ressources comme étant un appel. La capacité d'une cellule représenterait donc le nombre de canaux disponibles.

En se référant à l'équation 4.1 pour la charge ρ, on obtient l'expression mathématique suivante pour le calcul de la probabilité de blocage :

$$P_b = \frac{\frac{\rho^C}{C!}}{\sum_{x=0}^{C} \frac{\rho^x}{x!}} \tag{4.2}$$

Probabilité d'interruption forcée

Dans un premier temps, nous avons supposé que l'architecture du réseau d'accès est F-HMIPv6 qui est basé sur un modèle *break-before-make*. Ensuite, nous avons émis l'hypothèse que le réseau a atteint un état de régime permanent. De plus, nous avons défini notre réseau d'accès de telle sorte qu'un MN quittant le réseau par une arrête est immédiatement remplacé par un nouveau MN dans la cellule opposée. Ainsi, le nombre MN dans chaque cellule du réseau d'accès demeure statistiquement stable.

Dans un autre ordre d'idée, le protocole HPMRSVP-TE n'effectue aucune réservation de ressources à l'avance, comme cela est fait dans le cas de MRSVP.

En conséquence, la probabilité qu'une réservation de ressources soit refusée suite à une relève est identique à probabilité que la même réservation de ressources soit refusée à l'initiation de la connexion :

$$P_f = P_b \tag{4.3}$$

Probabilité de compléter une session avec succès

La probabilité de compléter une session indique la proportion des sessions qui ne sont pas bloquées initialement ou qui ne sont pas annulées lorsqu'une relève ne peut

pas être effectuée, faute de ressources disponibles.

Soit $N \geq 0$ le nombre de relèves subies pendant la durée d'une session temps-réel. En se référant aux équations 4.2 et 4.3, on exprime la probabilité de compléter une session de la façon suivante :

$$\begin{aligned} P_c &= (1 - P_b) \times (1 - P_f)^N \\ &= (1 - P_b)^{N+1} \end{aligned} \tag{4.4}$$

4.2 Étude des coûts de la solution

Dans cette section, nous procéderons à une étude des délais ainsi que du nombre de paquets échangés lors d'opérations courantes. Les opérations évaluées sont la réservation initiale des ressources, la relève intra-domaine sans bicasting et les rafraîchissements d'états de chemin et de réservation. Le réseau étudié est présenté à la Figure 4.2.

4.2.1 Hypothèses préalables

Avant de procéder à l'estimation des coûts reliés au protocole HPMRSVP-TE, il convient d'émettre des hypothèses qui nous permettront de simplifier l'étude du modèle :

- MN1 est émetteur et MN2 est récepteur ;
- MN1 et MN2 supportent toutes les fonctionnalités requises ;
- les messages échangés sont tous de la même taille ;
- les délais de traitement des routeurs et des MN sont négligeables ;
- les délais inter-domaines et du lien radio sont plus importants que les délais du réseau d'accès ;
- un mécanisme de signalisation interne à l'application est utilisé.

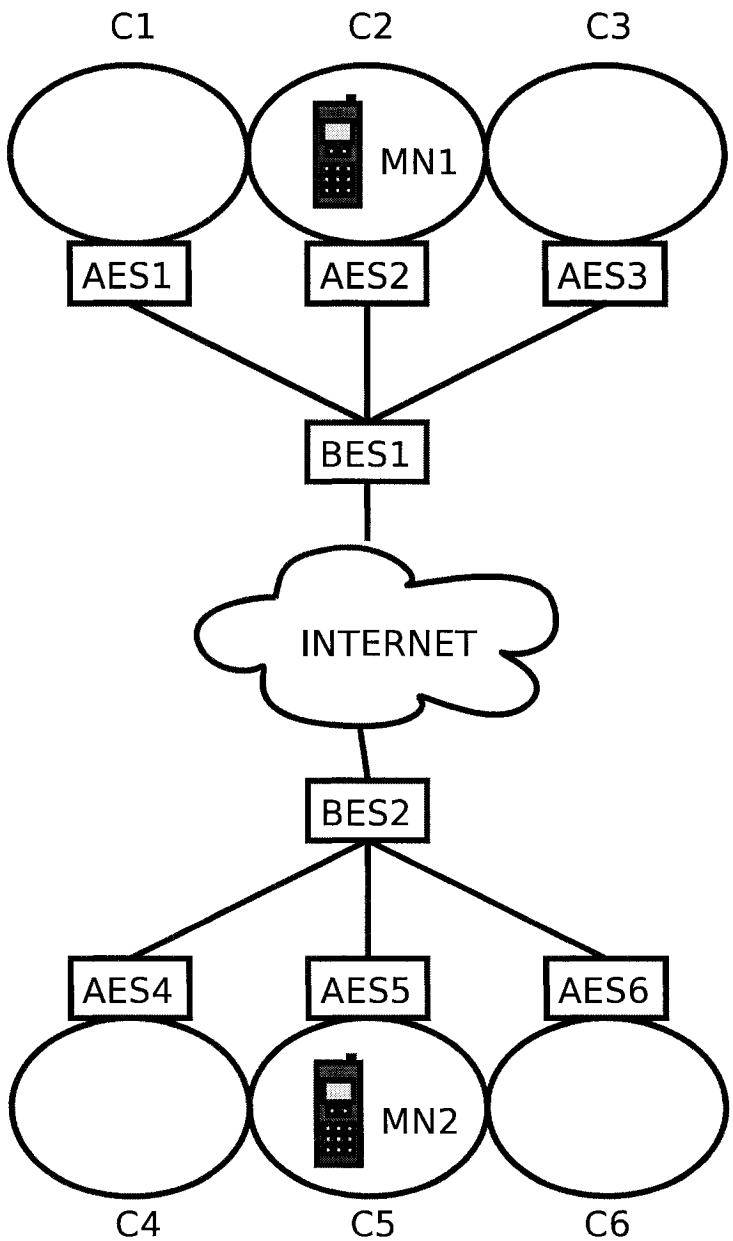

FIGURE 4.2 Réseau utilisé pour l'étude des coûts

4.2.2 Constantes du modèle

Avant de dévoiler les expressions mathématiques qui représentent les caractéristiques du modèle, il convient de définir les constantes de ce dernier. Le Tableau 4.2 présente les délais de transmission définis pour l'étude des coûts. Les délais de propagation sont considérés négligeables à l'intérieur du réseau d'accès.

TABLEAU 4.2 Délais évalués dans l'étude de coûts

Constante	Description
ΔT_{net}	Délai de transmission pour les liens AES/BES
ΔT_{rad}	Délai de transmission pour les liens radio
ΔT_{wan}	Délais de transmission et de propagation pour traverser l'Internet
ΔT_{E2E}	Délais de transmission et de propagation de bout-en-bout
ΔT_{L2HO}	Délai encouru pour effectuer une relève de niveau 2

4.2.3 Réservation initiale des ressources

Afin de fixer le début de notre analyse dans le temps, nous supposons qu'il s'agit du moment où MN1 envoie un message à MN2 décrivant le débit de la connexion ainsi que le `Flow Label` utilisé. Sur réception de ce message, MN2 confirmera qu'il accepte ce flot de données.

Le délai de bout-en-bout est calculé à l'aide de l'expression suivante :

$$\begin{aligned} \Delta T_{E2E} &= \Delta T_{rad} + \Delta T_{net} + \Delta T_{wan} + \Delta T_{net} + \Delta T_{rad} \\ &= 2(\Delta T_{rad} + \Delta T_{net}) + \Delta T_{wan} \end{aligned} \tag{4.5}$$

Avec HPMRSVP-TE, MN1 peut réserver des ressources sur le lien montant tandis que la description du flot de données transite vers MN2. Cette stratégie peut être mise à profit si les probabilités de blocage et de refus de la part de MN2, sont faibles.

MN1 envoie donc un message *Path* à BES1, qui répondra par un message *Resv*. La pénalité associée à cette réservation est comprise dans le processus global. En effet, la réservation de ressources dans le réseau d'accès de MN1 est complétée avant même que la description de flot soit rendue à MN2. On constate la relation suivante :

$$\Delta T_{init,sender} = 2(\Delta T_{rad} + \Delta T_{net}) < \Delta T_{E2E} \tag{4.6}$$

Lorsque MN2 reçoit la description de flot, ce dernier envoie un message *PathMod* à BES2 afin de lui donner les paramètres de QdS sur le lien descendant. BES2 émet ensuite un message *Path* à MN2 qui répond avec un message *Resv*, sans demander de confirmation de réservation. L'expression suivante donne le temps requis pour effectuer la réservation sur le lien descendant, dans le réseau d'accès de MN2 :

$$\Delta T_{init,receiver} = 3(\Delta T_{rad} + \Delta T_{net}) \tag{4.7}$$

Afin de réduire les délais avant l'envoi des premiers paquets de données par MN1, il est possible de tirer profit de la même stratégie que lors de l'établissement de la réservation sur le lien montant, si la probabilité de blocage le permet. En effet, le message d'acceptation de la description de flot peut être envoyé à MN1 aussitôt que MN2 a émis un message *PathMod* à l'intention de BES2. La réservation de ressources dans le réseau d'accès de MN2 est complétée avant que MN1 ait reçu la confirmation de ce premier, si la relation suivante peut être vérifiée :

$$\Delta T_{E2E} > 3(\Delta T_{rad} + \Delta T_{net}) \Longleftrightarrow \Delta T_{wan} > (\Delta T_{rad} + \Delta T_{net}) \tag{4.8}$$

Enfin, l'expression suivante indique le moment à partir duquel les réservations de ressources sont en force :

$$\Delta T_{init} = \begin{cases} 2\Delta T_{E2E} & \Delta T_{wan} > (\Delta T_{rad} + \Delta T_{net}) \\ \Delta T_{E2E} + 3(\Delta T_{rad} + \Delta T_{net}) & \text{sinon} \end{cases} \tag{4.9}$$

Pour conclure, MN1 peut commencer à émettre des paquets de données aussitôt qu'il a reçu la confirmation de MN2 et ce même si $\Delta T_{wan} < (\Delta T_{rad} + \Delta T_{net})$. En effet, les délais encourus pour traverser le réseau d'accès de MN1 ainsi que l'Internet dépassent largement le temps manquant pour compléter la réservation initiale.

4.2.4 Modification d'une réservation de ressources

Pour le cas de modification d'une réservation existante, nous avons déterminé que l'expression des coûts est identique à celle d'une réservation initiale. En conséquence, en se basant sur (4.9), on obtient l'expression suivante :

$$\Delta T_{modif} \equiv \Delta T_{init} \tag{4.10}$$

4.2.5 Rafraîchissement des états de réservation

Le rafraîchissement des états de chemin et de réservation s'effectue par l'échange de messages *Path* et *Resv* entre les entités concernées. Ce processus doit être répété à intervalles réguliers pour conserver active une réservation de ressources. Ainsi, si les entités négligent de détruire explicitement une réservation après usage, cette dernière sera éventuellement démantelée automatiquement par le réseau. Le but de ce mécanisme est d'éviter une congestion du réseau due à des pannes ou de la négligence.

Le protocole HPMRSVP-TE a pour particularité d'économiser les ressources radio en évitant au MN d'effectuer lui-même les traitements liés aux rafraîchissements des états de chemin et de réservation. En effet, ce rôle est octroyé au AES qui rafraîchit ces états au nom du MN. On peut procéder ainsi puisque le AES dispose d'informations sur l'état du MN par l'entremise de la couche 2.

Dans un autre ordre d'idée, les messages de rafraîchissement se limitent au réseau d'accès en évitant la propagation de bout-en-bout. Il est important de mentionner que les rafraîchissements des états de chemin et de réservation s'effectuent de façon indépendante dans les deux réseaux d'accès de notre modèle.

L'expression suivante représente donc les coûts engendrés pour le rafraîchissement des états de chemin et de réservation, pour un seul réseau d'accès, avec ou sans implication du MN :

$$\Delta T_{refresh} = \begin{cases} 2\Delta T_{net} & \text{sans implication du MN} \\ 2(\Delta T_{rad} + \Delta T_{net}) & \text{avec implication du MN} \end{cases} \tag{4.11}$$

Il est important de mentionner que les échanges de messages de rafraîchissement avec implication du MN sont rares puisque la détection de la présence de ce dernier est assurée par des mécanismes de la couche 2. Par contre, ils peuvent servir à la détection d'erreurs de fonctionnement du MN.

4.2.6 Relève intra-domaine sans bicasting

L'implémentation d'un réseau dérivé de l'architecture HMIPv6 permet à un MN d'effectuer une relève intra-domaine, sous l'autorité d'un MAP, tout en conservant sa RCoA. Ceci a pour conséquence que les relèves intra-domaines n'ont aucun impact sur le CN, à moins qu'un changement de MAP n'implique le changement de RCoA.

Pour le protocole HPMRSVP-TE, il faut distinguer les cas où la relève est effectuée par l'émetteur ou par le récepteur. En effet, dans le cas du récepteur, le BES peut envoyer le message *Path* qui décrit les paramètres de QdS du lien descendant pendant que le MN effectue la relève de niveau 2.

Enfin, les opérations qui constituent une relève intra-domaine sans bicasting sont présentées à la Figure 3.24. Toutefois, il est important de ne considérer que les délais pendant lesquels la QdS est affectée.

Relève effectuée par un émetteur

Le temps requis pour qu'un émetteur effectue une relève est compté à partir du moment de l'envoi d'un message *PathTear* vers le BES. En effet, ce message indique au AES que la réservation de ressources sur le lien montant n'est plus nécessaire. Ensuite, le MN procède à la relève de niveau 2 puis signale sa présence au NAES par l'envoi d'un message *FNA*. Ce message déclenche la transmission de tous les paquets accumulés par le NAES. Enfin, le MN établit la réservation sur le lien montant par un échange de messages *Path* et *Resv* avec le BES.

L'expression du délai total pour la relève effectuée par un émetteur est :

$$\begin{aligned} \Delta T_{handover,sender} &= \Delta T_{rad} + \Delta T_{L2HO} + \Delta T_{rad} + 2(\Delta T_{net} + \Delta T_{rad}) \\ &= 4\Delta T_{rad} + 2\Delta T_{net} + \Delta T_{L2HO} \end{aligned} \tag{4.12}$$

Relève effectuée par un récepteur

Le temps requis pour qu'un récepteur effectue une relève est compté à partir du moment de l'envoi d'un message *ResvTear* vers le BES. En effet, ce message indique au AES que la réservation de ressources sur le lien descendant n'est plus nécessaire.

Afin de décrire les paramètres de QdS du nouveau lien descendant, le BES envoie un message *Path* à la nouvelle adresse LCoA du MN suite à l'émission du message *FBACK*. Le message *Path* installe les états de chemins dans le NAES. Le message *Path* que le NAES destine au MN est alors accumulé en attendant que ce dernier se manifeste.

Pendant ce temps. le MN détruit les réservations existantes puis procède à la relève de niveau 2. Aussitôt reconnecté, le MN déclenche la transmission de tous les paquets accumulés par l'envoi d'un message *FNA* au NAES.

Le message *Path* décrivant les états de chemin du lien descendant est alors reçu et traité par le MN. Ce dernier complète la réservation des ressources par l'envoi d'un message *Resv*.

L'expression du délai total pour la relève effectuée par un récepteur est :

$$\begin{aligned} \Delta T_{handover,receiver} &= \Delta T_{rad} + \Delta T_{L2HO} + 2\Delta T_{rad} + (\Delta T_{net} + \Delta T_{rad}) \\ &= 4\Delta T_{rad} + \Delta T_{net} + \Delta T_{L2HO} \end{aligned} \tag{4.13}$$

4.2.7 Discussion des résultats de l'étude de coûts

Scénario de réservation initiale de ressources

Lorsque le mécanisme de signalisation est interne à l'application, HPMRSVP-TE complète la réservation initiale de ressources aussi rapidement qu'un protocole, tel que MRSVP, qui nécessite l'envoi de messages *Path* et *Resv* de bout-en-bout. Ceci est dû au fait que l'émetteur attend une confirmation d'acceptation des paramètres de QdS de la part du récepteur.

En contrepartie, notre solution serait plus rapide que MRSVP si l'on utilisait un protocole de signalisation tel que SIP. En effet, exception faite de l'échange des messages de signalisation, HPMRSVP-TE évite tout échange de message de bout-en-bout, ce qui constitue un avantage notable si ΔT_{E2E} est important[1].

Scénario de modification d'une réservation existante

Nous avons déterminé que les expressions des coûts pour le scénario de modification de réservation sont respectivement identiques à celles des réservations initiales.

Scénario de rafraîchissement des états de chemin et de réservation

Notre solution limite l'échange des messages de rafraîchissement à un seul réseau d'accès. De plus, en évitant d'impliquer le MN dans le processus de rafraîchissement, on réduit les besoins en bande passante et élimine le gaspillage des ressources radio.

[1] L'avantage découle du fait que l'on peut raisonnablement supposer que $\Delta T_{wan} \gg \Delta T_{net}$.

Scénarios de relèves intra-domaines

Lors des relèves intra-domaines, les délais encourus pendant lesquels la QdS est affectée sont principalement causés par la nature même de l'architecture F-HMIPv6 qui est basée sur un modèle *break-before-make*.

En effet, F-HMIPv6 minimise le risque de perte de paquet en accumulant, dans le NAES, les paquets destinés au MN qui sont reçus pendant que celui-ci effectue une relève de niveau 2. L'accumulation des paquets provoque une interruption temporaire du débit de données vers le MN.

De plus, le MN doit s'abstenir d'émettre des paquets pendant la relève de niveau 2, ce qui cause aussi une interruption temporaire du débit de données vers le CN.

Enfin, il faut considérer le temps requis pour rétablir les réservations de ressources une fois le MN connecté au NAES. Pendant ce délai, les paquets émis par le MN se voient temporairement assignés à la classe de service *best-effort*. En conséquence, le débit de données dans les deux directions est affecté pendant la durée de la relève.

En conclusion, une implémentation du protocole HPMRSVP-TE aurait avantage à être déployée à l'intérieur d'un réseau d'accès basé sur un modèle *make-before-break* afin d'offrir aux usagers une QdS constante, et ce même lors des relèves. En ayant la capacité de réserver des ressources via le NAES tout en conservant intactes les réservations existantes avec le PAES, le MN aurait la possibilité de migrer toutes ses connexions vers le NAES avant de se déconnecter du PAES, assurant ainsi une relève transparente pour l'usager.

4.3 Revue du processus de validation

Dans un premier temps, nous avons évalué les probabilités de blocage, d'interruption forcée ainsi que d'achèvement avec succès d'une session pour HPMRSVP-TE.

Par la suite, nous avons effectué une étude de coûts basée sur les délais et l'utilisation des ressources réseau. Pour ce faire, nous avons mis sur pied un environnement de test pour lequel nous avons évalué quatre scénarios d'opérations courantes pour le protocole HPMRSVP-TE. Les scénarios évalués sont :

- la réservation initiale de ressources ;
- la modification d'une réservation existante ;
- le rafraîchissement des états de chemin et de réservation ;

– la relève intra-domaine sans bicasting.

L'étude de coûts nous a permis de souligner que HPMRSVP-TE tire profit de la limitation des échanges de messages au réseau d'accès ainsi que d'un mécanisme optimisé de rafraîchissement qui minimise l'utilisation du lien radio.

En contrepartie, l'architecture F-HMIPv6 qui a servi de référence est basée sur un modèle *break-before-make.* En conséquence, on note une dégradation temporaire de la QdS pendant la relève de niveau 3 puisque les réservations de ressources ne sont pas établies préalablement à la connexion du MN au NAES. Toutefois, ce problème disparaîtra de lui-même lorsqu'une nouvelle architecture *make-before-break* sera mise en place.

CHAPITRE 5

CONCLUSION

Dans ce chapitre, nous réviserons les points importants du projet de recherche. Dans un premier temps, nous avons dressé une liste exhaustive des paradigmes de QdS existants dans la littérature pour le réseau Internet.

Dans le premier volet de notre proposition de solution, nous avons suggéré l'utilisation de l'architecture IPngLS à l'intérieur des limites du réseau d'accès.

Le second volet de notre proposition de solution consistait en l'élaboration d'un protocole de réservation de ressources et de distribution d'étiquettes. Ce protocole est fortement inspiré de RSVP-TE ainsi que de HPMRSVP.

La validation de l'approche proposée s'est effectuée en deux parties. Tout d'abord, nous avons construit un modèle analytique afin d'évaluer les probabilités de blocage, d'interruption forcée d'une session ainsi que d'achèvement d'une session avec succès.

La seconde partie du processus a consisté en une étude de coûts en ressources. Nous avons donc comptabilisé les échanges de messages entre les entités d'un réseau et évalué les temps requis pour effectuer quatre scénarios prédéterminés.

L'étude de coûts a souligné les avantages significatifs de HPMRSVP-TE. Toutefois, nous avons noté une dégradation de la QdS pendant la relève intra-domaine qui est causée par l'architecture F-HMIPv6 sous-jacente. En conséquence, lorsque l'architecture F-HMIPv6 sera remplacée par une autre basée sur un modèle *make-before-break*, les relèves deviendront transparentes pour l'usager.

Ce chapitre vient clore les démarches entreprises dans notre projet. Nous ferons une synthèse des travaux accomplis et présenterons l'ensemble des limitations de HPMRSVP-TE. Nous traiterons des problèmes de dégradation de la QdS qui, lors des relèves intra-domaines, peuvent survenir malgré le recours à l'architecture F-HMIPv6. Enfin, nous discuterons du support du multicasting qui est actuellement inexistant.

5.1 Synthèse des travaux

L'une des contributions de ce travail de recherche est de proposer l'utilisation de l'architecture IPngLS à l'intérieur du réseau d'accès. En conséquence, il devient possible de construire des chemins commutés par étiquette. On tire ainsi profit des mécanismes existants dans l'architecture MPLS mais sans devoir l'utiliser. Cette méthode a l'avantage de simplifier la pile de protocoles.

Une seconde contribution de ce mémoire, beaucoup plus importante cette fois, est l'adaptation de RSVP-TE, pour la réservation des ressources ainsi que la distribution des étiquettes, à l'intérieur d'un réseau d'accès cellulaire. De plus, nous avons greffé à RSVP-TE les caractéristiques les plus significatives du protocole HPMRSVP :

- le traitement des messages de rafraîchissement par le AES, au nom du MN ;
- la limitation des réservations de ressources au réseau d'accès seulement ;
- les requêtes de réservations de ressources pour le lien descendant sont effectuées par le BES, au nom du CN, pour des raisons de sécurité.

L'originalité de la solution proposée dans ce mémoire réside en la modification de l'objet `SESSION` de RSVP-TE afin de remédier au problème de propagation du champ `Flow Label` de IPv6. Cette modification constitue une solution élégante qui permet au BES et aux AES de récupérer la valeur originelle du `Flow Label`, assurant ainsi la transparence du réseau d'accès et le respect des normes de l'IETF.

5.2 Limitations de la solution proposée

Au terme de notre projet, nous avons atteint les objectifs de recherche que nous avions fixés au début de celui-ci. En effet, l'utilisation de HPMRSVP-TE dans le réseau d'accès permet de déplacer les réservations de ressources et les chemins commutés par étiquette. De plus, l'utilisation de l'architecture F-HMIPv6 réduit considérablement les risques de perte de paquets lors de la relève intra-domaine. Aussi, l'utilisation de l'architecture IPngLS tire profit du champ `Flow Label` du protocole IPv6 pour la création de LSP. Enfin, HPMRSVP-TE solutionne le problème de propagation du `Flow Label`, assurant ainsi la transparence du réseau d'accès.

Nous avons volontairement restreint la portée de notre projet de recherche afin de nous concentrer sur les objectifs que nous croyons les plus importants. En conséquence, les points abordés dans les sous-sections suivantes sont laissés en travaux futurs.

5.2.1 Variété des technologies d'accès

Nous avons volontairement restreint notre projet de recherche aux technologies d'accès cellulaires en général. En effet, ces dernières possèdent des mécanismes de QdS permettant de réserver des ressources sous forme de tranches de temps. De plus, ces technologies ont depuis longtemps assuré des relèves de niveau 2 qui soient transparentes pour les usagers.

Enfin, nous avons omis de considérer des technologies d'accès sans-fil telles que *WLAN* (IEEE 802.11), *Bluetooth* (IEEE 802.15) et *WiMax* (IEEE 802.16). Il est clair que la technologie d'accès employée affectera la capacité de HPMRSVP-TE à garantir les réservations de ressources sur le lien radio.

5.2.2 Dégradation de la QdS lors des relèves intra-domaines

Le choix de l'architecture F-HMIPv6 a permis de réduire considérablement les risques de perte de paquets lors des relèves intra-domaines. Par contre, nous n'avons pas éliminé complètement les risques de dégradation de la QdS, puisque le modèle de relève est basé sur le concept *break-before-make*.

Possibilité de blocage lors de la relève

La première conséquence à l'utilisation du modèle *break-before-make* est la possibilité de blocage qui résulterait en l'incapacité de réserver des ressources au niveau du NAES, et ce une fois la relève enclenchée. La conséquence d'un blocage serait la perte subite de QdS puisque pour toutes les réservations étant refusées, les paquets concernés seraient insérés dans la classe de service *best-effort*.

Retard de paquets appartenant à des flots temps-réel

Une autre conséquence à l'utilisation du modèle *break-before-make* est le retard de certains paquets appartenant à des sessions temps-réel. En effet, F-HMIPv6 élimine presque totalement le risque de perte de paquets pendant la relève puisque les paquets destinés au MN sont accumulés dans le NAES. Toutefois, ce délai d'attente peut être suffisant pour rendre désuets certains paquets temps-réel accumulés.

Dans un autre ordre d'idée, il y a, au moment de la reconnexion, une courte période de vulnérabilité pendant laquelle les réservations de ressources ne sont pas

encore complétées. Les paquets destinés au MN serait alors temporairement associés à la classe de service *best-effort*. En conséquence, des paquets destinés au MN pourraient être perdus lors d'une situation de forte congestion.

5.2.3 Transfert du contexte matériel lors de la relève

La solution proposée dans ce mémoire n'adresse pas le transfert de contexte entre les AES, lors des relèves. Il serait préférable de considérer ce sujet lors d'études visant à améliorer les mécanismes de relève au niveau 2. De plus, le problème deviendra encore plus complexe à résoudre lorsque l'on considèrera les relèves inter-technologies.

Parmi les informations qui font partie du contexte, on retrouve la compression d'en-têtes de protocoles. Dans les années 90, ces sujets ont été abordés pour les liens sériels de faible capacité (Jacobson, 1990; Degermark *et al.*, 1999; Casner et Jacobson, 1999). Par contre, ces mécanismes sont peu efficaces pour des liens dont le taux d'erreur et le délai de réponse sont élevés. Toutefois, la nécessité d'économiser les ressources radio et la tendance à migrer tous les services vers IP ravive les intérêts dans ce domaine (Bormann *et al.*, 2001).

5.2.4 Relève intra-domaine avec changement de MAP

Les relèves intra-domaines impliquant un changement de MAP n'ont pas été considérées dans notre projet. Ce cas particulier a été traité par Soliman *et al.* (2005). Toutefois, dans le cas des réservations de ressources, le nouveau MAP ne possède pas les informations pour le lien descendant. Cette situation forcerait le MN à lui spécifier les paramètres de QdS à l'aide d'un message *PathMod.*

5.2.5 Support des relèves inter-domaines

Lorsque nous avons énoncé les objectifs de recherche de notre projet, nous avons affirmé ne pas considérer les relèves inter-domaines puisqu'elles sont plus difficiles à réaliser que les relèves intra-domaines.

La complexité supplémentaire due aux relèves inter-domaines provient de l'interaction entre deux réseaux appartenant à des opérateurs différents. Cette situation nécessite donc des ententes entre les fournisseurs pour la gestion de la facturation et l'établissement de réservations de ressources.

La gestion de ces ententes de services déborde largement du cadre de ce mémoire. En effet, les difficultés inhérentes aux relèves inter-domaines sont principalement d'ordre administratif et incorporent un grand volet voué à la sécurité.

Enfin, la plus grande difficulté technique, hormis les aspects administratifs et de sécurité, est que les relèves inter-domaines impliquent deux BES alors que les relèves intra-domaines se limitent souvent au domaine d'un seul BES.

5.3 Améliorations futures

Cette section présente les avenues où des améliorations pourraient être apportées au protocole HPMRSVP-TE. Contrairement aux limitations qui furent décrites dans la section 5.2 et qui posent problème pour l'utilisation à grande échelle du protocole HPMRSVP-TE, les améliorations décrites ci-dessous constituent des ajouts concrets de fonctionnalités.

5.3.1 RFC2961 – Refresh Overhead Reduction Extensions

Il serait souhaitable d'incorporer au protocole HPMRSVP-TE les mécanismes de réduction de la charge de traitement tels que définis par Berger *et al.* (2001).

L'une des fonctionnalités proposée est le message `Bundle` qui permet de regrouper plusieurs messages RSVP en un seul. Le message `Bundle` serait particulièrement utile lors des relèves, alors que les ressources appartenant à plusieurs sessions sur le chemin du PAES doivent être libérées simultanément. Ainsi, l'envoi d'un seul message regrouperait tous les messages *PathTear* et *ResvTear*, ce qui aurait comme impact de diminuer la charge de traitement des routeurs impliqués.

Un autre mécanisme défini par Berger *et al.* (2001) est la création d'un objet servant à identifier un message complet. Ainsi, il serait possible de réduire considérablement la taille des messages échangés entre les nœuds du réseau. Cette caractéristique est particulièrement intéressante pour les messages de rafraîchissement qui doivent répéter, à intervalles réguliers, les mêmes informations afin de maintenir les états de réservations.

5.3.2 RFC4090 – Fast Reroute Extensions

Il serait possible d'intégrer au protocole HPMRSVP-TE les mécanismes de réparation locale définis pour RSVP-TE (Pan *et al.*, 2005). En effet, ces mécanismes permettraient de contourner automatiquement un point de défaillance, que ce soit un nœud ou un lien. De plus, le risque de dégradation de la QdS s'en trouverait amoindri puisque les ressources seraient déjà réservées sur les liens de contournement.

5.3.3 Support du multicasting

Dans sa version actuelle, HPMRSVP-TE ne supporte pas les sessions *multicast*, tout comme RSVP-TE dont il s'inspire directement. Toutefois, puisque nous avons conçu HPMRSVP-TE orienté récepteur, nous n'introduisons pas d'incompatibilité avec le support éventuel du *multicasting*.

5.3.4 ARTP – *Access Router Tunneling Protocol*

Une façon de réduire les impacts d'une relève sur la QdS serait d'incorporer le protocole proposé par Zhang *et al.* (2005). En effet, ce protocole permet l'établissement de tunnels, pouvant être bidirectionnels, entre le PAR et le NAR.

Une approche basée sur l'établissement d'un tunnel entre les AR permettrait le transfert du contexte matériel. Nous avons mentionné plus tôt que les informations relatives à la compression d'en-tête devraient être chargées dans le NAR. En conséquence, les identificateurs de connexions seraient conservés, évitant ainsi le transfert de nombreux paquets avec des en-têtes complets jusqu'à ce que de nouveaux identificateurs soient générés. Enfin, cette approche pourrait permettre de réduire au minimum la dégradation de la QdS lors de la relève de niveau 3.

5.3.5 Présence de deux modems dans les appareils mobiles

Les scénarios de relève présentés à la Figure 3.24 et à la Figure 3.25 introduisent une courte période de temps pendant laquelle il peut y avoir dégradation de la QdS. En effet, que les appareils mobiles d'aujourd'hui soient dotés d'un seul modem rend difficile la réservation de nouvelles ressources au niveau du NAES tant que le MN ne s'est pas déconnecté du PAES. Il en résulte un modèle *break-before-make*.

L'introduction d'un second modem à l'intérieur des appareils mobiles permettrait au MN d'être connecté à deux points d'accès simultanément, et ce pendant toute la durée de la relève. Il en résulterait ainsi un modèle *make-before-break*.

La Figure 5.1 montre ce que pourrait être une relève initiée par un MN doté de deux modems. Celui-ci possèderait alors la capacité de garantir que les nouvelles réservations de ressources soient mises en place entre le MN situé au NAES et le BES. Cette validation serait effectuée avant même de procéder à la déconnexion au niveau du PAES.

On peut remarquer que, dans la Figure 5.1, le moment où le MN se déconnecte du PAES est repoussé au-delà des réservations de ressources au niveau du NAES. Ainsi, le MN peut assurer une transition totalement transparente de toutes ses connexions. L'usager ne subit alors aucune dégradation de QdS. En conséquence, les mécanismes de F-HMIPv6 ne sont plus nécessaires.

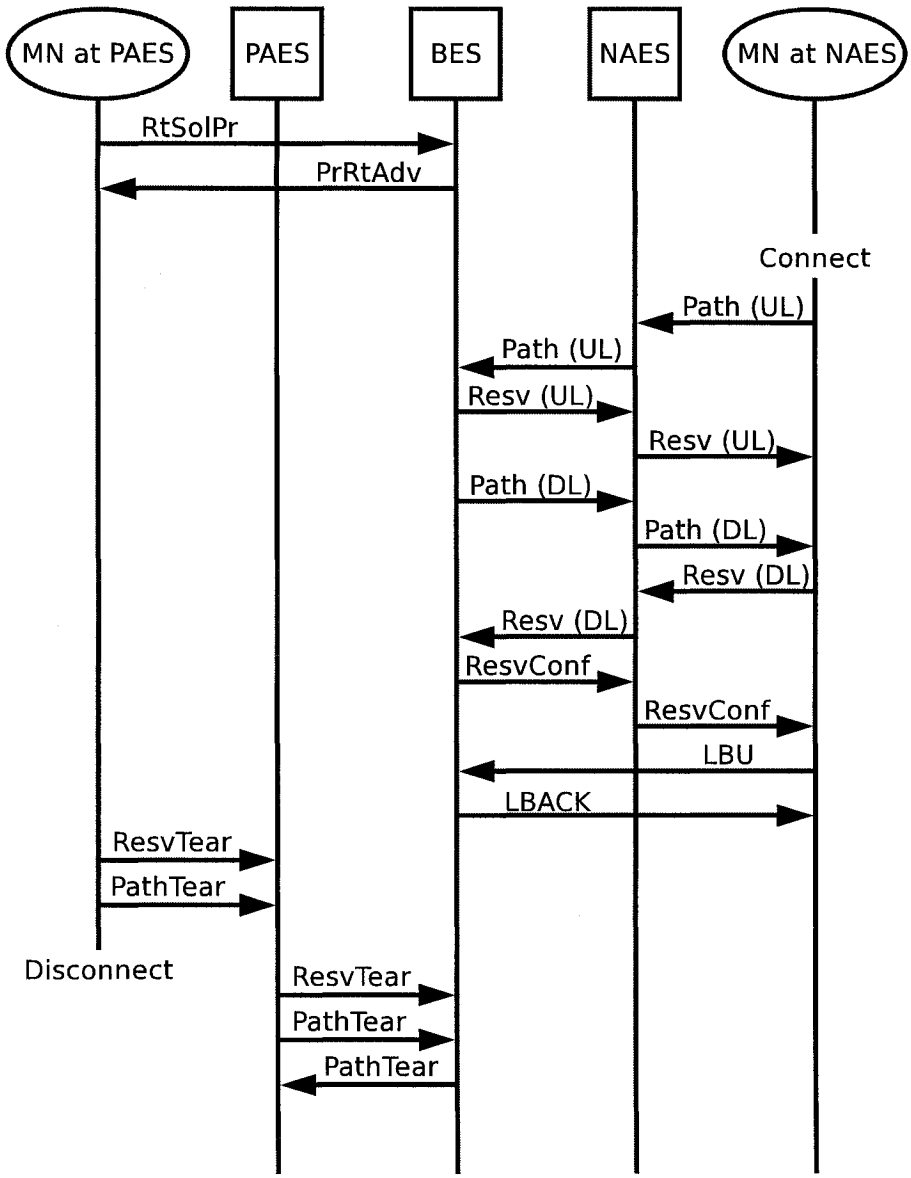

FIGURE 5.1 Relève initiée par un MN doté de deux modems

Références

ABONDO, C. (2005). *Gestion de la Qualité de Service dans les systèmes mobiles de prochaine génération.* Thèse de doctorat, École Polytechnique de Montréal.

ABONDO, C. ET PIERRE, S. (2004). *Hierarchical Proxy Mobile Resource Reservation Protocol.* IETF. Status : Internet DRAFT.

ALMQUIST, P. (1992). *RFC1349 : Type of Service in the Internet Protocol Suite.* IETF. Obsoleted by RFC2474.

ANDERSON, L. ET SWALLOW, G. (2003). *RFC3468 : The Multiprotocol Label Switching (MPLS) Working Group decision on MPLS signaling protocols.* IETF. Status : INFORMATIONAL.

ANDERSSON, L., DOOLAN, P., FELDMAN, N., FREDETTE, A. ET THOMAS, B. (2001). *RFC3036 : LDP Specification.* IETF. Status : STANDARDS TRACK.

ARKKO, J., KUIJPERS, G., SOLIMAN, H., LOUGHNEY, J. ET WILJAKKA, J. (2003). *RFC3316 : IPv6 for Some Second and Third Generation Cellular Hosts.* IETF. Status : INFORMATIONAL.

AWDUCHE, D., BERGER, L., GAN, D., LI, T., SRINIVASAN, V. ET SWALLOW, G. (2001a). *RFC3209 : RSVP-TE : Extensions to RSVP for LSP Tunnels.* IETF. Status : STANDARDS TRACK.

AWDUCHE, D., CHIU, A., ELWALID, A., WIDJAJA, I. ET XIAO, X. (2002). *RFC3272 : Overview and Principles of Internet Traffic Engineering.* IETF. Status : INFORMATIONAL.

AWDUCHE, D., HANNAN, A. ET XIAO, X. (2001b). *RFC3210 : Applicability Statement for Extensions to RSVP for LSP-Tunnels.* IETF. Status : INFORMATIONAL.

AWDUCHE, D., MALCOLM, J., AGOGBUA, J., O'DELL, M. ET MCMANUS, J. (1999). *RFC2702 : Requirements for Traffic Engineering Over MPLS.* IETF. Status : INFORMATIONAL.

BAKER, F., LINDELL, B. ET TALWAR, M. (2000). *RFC2747 : RSVP Cryptographic Authentication.* IETF. Status : STANDARDS TRACK.

BERGER, L., GAN, D., SWALLOW, G., PAN, P., TOMMASI, F. ET MOLENDINI, S. (2001). *RFC2961 : RSVP Refresh Overhead Reduction Extensions.* IETF. Status : STANDARDS TRACK.

BERNET, Y. (2000). *RFC2996 : Format of the RSVP DCLASS Object.* IETF. Status : STANDARDS TRACK.

BERNET, Y., FORD, P., YAVATKAR, R., BAKER, F., ZHANG, L., SPEER, M., BRADEN, R., DAVIE, B., WROCLAWSKI, J. ET FELSTAINE, E. (2000a). *RFC2998 : A Framework for Integrated Services Operation over Diffserv Networks.* IETF. Status : INFORMATIONAL.

BERNET, Y. ET PABBATI, R. (2000). *RFC2872 : Application and Sub Application Identity Policy Element for Use with RSVP.* IETF. Status : STANDARDS TRACK.

BERNET, Y., SMITH, A. ET DAVIE, B. (2000b). *RFC2997 : Specification of the Null Service Type.* IETF. Status : STANDARDS TRACK.

BLACK, D., BRIM, S., CARPENTER, B. ET LE FAUCHEUR, F. (2001). *RFC3140 : Per Hop Behavior Identification Codes.* IETF. Obsoletes RFC2836. Status : STANDARDS TRACK.

BLAKE, S., BLACK, D., CARLSON, M., DAVIES, E., WANG, Z. ET WEISS, W. (1998). *RFC2475 : An Architecture for Differentiated Services.* IETF. Status : INFORMATIONAL.

BORMANN, C., BURMEISTER, C., DEGERMARK, M., FUKUSHIMA, H., HANNU, H., JONSSON, L.-E., HAKENBERG, R., KOREN, T., LE, K., LIU, Z., MARTENSSON, A., MIYAZAKI, A., SVANBRO, K., WIEBKE, T., YOSHIMURA, T. ET ZHENG, H. (2001). *RFC3095 : RObust Header Compression (ROHC) : Framework and four profiles : RTP, UDP, ESP, and uncompressed.* IETF. Status : STANDARDS TRACK.

BRADEN, R., CLARK, D. ET SHENKER, S. (1994). *RFC1633 : Integrated Services in the Internet Architecture : an Overview.* IETF. Status : INFORMATIONAL.

BRADEN, R. ET ZHANG, L. (2001). *RFC3097 : RSVP Cryptographic Authentication – Updated Message Type Value.* IETF. Updates RFC2747. Status : STANDARDS TRACK.

BRADEN, R., ZHANG, L., BERSON, S., HERZOG, S. ET JAMIN, S. (1997). *RFC2205 : Resource ReSerVation Protocol (RSVP) Version 1 – Functional Specification.* IETF. Status : PROPOSED STANDARD.

BRADNER, S. (1997). *RFC2119 : Key Words to use in RFCs to Indicate Requirement Level.* IETF. Status : INFORMATIONAL.

CASNER, S. ET JACOBSON, V. (1999). *RFC2508 : Compressing IP/UDP/RTP Headers for Low-Speed Serial Links.* IETF. Status : STANDARDS TRACK.

CHAKRAVORTY, S. (2005). *IPv6 Label Switching Architecture.* IETF. Status : Internet DRAFT.

CHARNY, A., BENNETT, J., BENSON, K., LE BOUDEC, J., CHIU, A., COURTNEY, W., DAVARI, S., FIROIU, V., KALMANEK, C. ET RAMAKRISHNAN, K. (2002). *RFC3247 : Supplemental Information for the New Definition of the EF PHB (Expedited Forwarding Per-Hop Behavior).* IETF. Status : INFORMATIONAL.

CONTA, A. ET DEERING, S. (1998). *RFC2463 : Internet Control Message Protocol for IPv6.* IETF. Obsoletes RFC1885. Status : STANDARDS TRACK.

CROCKER, D. ET OVERELL, P. (1997). *RFC2234 : Augmented BNF for Syntax Specifications : ABNF.* IETF. Status : STANDARDS TRACK.

DAVIE, B., CHARNY, A., BENNETT, J., BENSON, K., LE BOUDEC, J., COURTNEY, W., DAVARI, S., FIROIU, V. ET STILIADIS, D. (2002). *RFC3246 : An Expedited Forwarding PHB (Per-Hop Behavior).* IETF. Obsoletes RFC2598. Status : STANDARDS TRACK.

DEERING, S. ET HINDEN, R. (1995). *RFC1883 : Internet Protocol, Version 6 (IPv6) Specification.* IETF. Obsoleted by RFC2460.

DEERING, S. ET HINDEN, R. (1998). *RFC2460 : Internet Protocol, Version 6 (IPv6) Specification.* IETF. Obsoletes RFC1883. Status : DRAFT STANDARD.

DEGERMARK, M., NORDGREN, B. ET PINK, S. (1999). *RFC2507 : IP Header Compression.* IETF. Status : STANDARDS TRACK.

ELLEINGAND, S. (2004). *Mécanisme FH-RSVP pour la relève intrasite dans les réseaux hiérarchiques Mobile IPv6.* Mémoire de maîtrise, École Polytechnique de Montréal.

GROSSMAN, D. (2002). *RFC3260 : New Terminology and Clarifications for Diffserv.* IETF. Updates RFC2474, RFC2475 and RFC2597. Status : INFORMATIONAL.

HAMER, L.-N., GAGE, B., KOSINSKI, B. ET SHIEH, H. (2003). *RFC3520 : Session Authorization Policy Element.* IETF. Status : STANDARDS TRACK.

HANDLEY, M. ET JACOBSON, V. (1998). *RFC2327 : SDP : Session Description Protocol.* IETF. Status : STANDARDS TRACK.

HEINANEN, J., FINLAND, T., BAKER, F., WEISS, W. ET WROCLAWSKI, J. (1999). *RFC2597 : Assured Forwarding PHB Group.* IETF. Status : STANDARDS TRACK. Updated by RFC3260.

HERZOG, S. (2000). *RFC2750 : RSVP Extensions for Policy Control.* IETF. Updates RFC2205. Status : PROPOSED STANDARD.

HERZOG, S. (2001). *RFC3181 : Signaled Preemption Priority Policy Element.* IETF. Obsoletes RFC2751. Status : STANDARDS TRACK.

JACOBSON, V. (1990). *RFC1144 : Compressing TCP/IP Headers for Low-Speed Serial Links.* IETF. Status : STANDARDS TRACK.

JACOBSON, V., NICHOLS, K. ET PODURI, K. (1999). *RFC2598 : An Expedited Forwarding PHB.* IETF. Obsoleted by RFC3246.

JAMOUSSI, B., ANDERSSON, L., CALLON, R., DANTU, R., WU, L., DOOLAN, P., WORSTER, T., FELDMAN, N., FREDETTE, A., GIRISH, M., GRAY, E., HEINANEN, J., KILTY, T. ET MALIS, A. (2002). *RFC3212 : Constraint-Based LSP Setup using LDP.* IETF. Status : STANDARDS TRACK.

JHA, S. ET HASSAN, M. (2002). *Engineering Internet QoS.* Artech House.

JOHNSON, D., PERKINS, C. ET ARKKO, J. (2004). *RFC3775 : Mobility Support in IPv6.* IETF. Status : STANDARDS TRACK.

JUNG, H., SOLIMAN, H., JOO KOH, S. ET TAKAMIYA, N. (2005). *Fast Handover for Hierarchical MIPv6 (F-HMIPv6).* IETF. Status : Internet DRAFT.

KOMPELLA, K. ET LANG, J. (2004). *RFC3936 : Procedures for Modifying the Resource reSerVation Protocol (RSVP).* IETF. Updates RFC2205 and RFC3209. Status : Best Current Practices.

KOODLI, R. (2005). *RFC4068 : Fast Handovers for Mobile IPv6.* IETF. Status : EXPERIMENTAL.

LI, T. ET REKHTER, Y. (1998). *RFC2430 : A Provider Architecture for Differentiated Services and Traffic Engineering (PASTE).* IETF. Status : INFORMATIONAL.

NARTEN, T., NORDMARK, E. ET SIMPSON, W. (1998). *RFC2461 : Neighbor Discovery for IPv6.* IETF. Obsoletes RFC1970. Status : STANDARDS TRACK.

NICHOLS, K., BLAKE, S., BAKER, F. ET BLACK, D. (1998). *RFC2474 : Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers.* IETF. Obsoletes RFC1349 and RFC1455. Status : STANDARDS TRACK.

NICHOLS, K. ET CARPENTER, B. (2001). *RFC3086 : Definition of Differentiated Services Per Domain Behaviors and Rules for their Specification.* IETF. Status : INFORMATIONAL.

OLSON, S., CAMARILLO, G. ET B. ROACH, A. (2002). *RFC3266 : Support for IPv6 in Session Description Protocol (SDP).* IETF. Status : STANDARDS TRACK.

PAN, P., SWALLOW, G. ET ATLAS, A. (2005). *RFC4090 : Fast Reroute Extensions to RSVP-TE for LSP Tunnels.* IETF. Status : STANDARDS TRACK.

PARTRIDGE, C. (1995). *RFC1809 : Using the Flow Label Field in IPv6.* IETF. Status : INFORMATIONAL.

PARTRIDGE, C. ET JACKSON, A. (1999). *RFC2711 : IPv6 Router Alert Option.* IETF. Status : STANDARDS TRACK.

RAJAHALME, J., CONTA, A., CARPENTER, B. ET DEERING, S. (2004). *RFC3697 : IPv6 Flow Label Specification.* IETF. Status : STANDARDS TRACK.

RAMAKRISHNAN, K., FLOYD, S. ET BLACK, D. (2001). *RFC3168 : The Addition of Explicit Congestion Notification (ECN) to IP.* IETF. Obsoletes RFC2481. Updates RFC793, RFC2401 and RFC2474. Status : STANDARDS TRACK.

REYNOLDS, J. (2002). *RFC3232 : Assigned Numbers : RFC 1700 is Replaced by an On-line Database.* IETF. Status : INFORMATIONAL.

ROESLER, V., BALBINOT, L., DE ANDRADE, M. ET TAROUCO, L. (2002). IP next generation label switching. *IEEE Workshop on IP Operations and Management.*

ROSEN, E., TAPPAN, D., FEDORKOW, G., REKHTER, Y., FARINACCI, D., LI, T. ET CONTA, A. (2001a). *RFC3032 : MPLS Label Stack Encoding.* IETF. Status : STANDARDS TRACK.

ROSEN, E., VISWANATHAN, A. ET CALLON, R. (2001b). *RFC3031 : Multiprotocol Label Switching Architecture.* IETF. Status : STANDARDS TRACK.

ROSENBERG, J., SCHULZRINNE, H., CAMARILLO, G., JOHNSTON, A., PETERSON, J., SPARKS, R., HANDLEY, M. ET SCHOOLER, E. (2002). *RFC3261 : Session Initiation Protocol (SIP).* IETF. Status : PROPOSED STANDARD.

SCHULZRINNE, H., CASNER, S., FREDERICK, R. ET JACOBSON, V. (2003). *RFC3550 : RTP : A Transport Protocol for Real-Time Applications.* IETF. Status : STANDARD.

SHENKER, S., PARTRIDGE, C. ET GUERIN, R. (1997). *RFC2212 : Specification of Guaranteed Quality of Service.* IETF. Status : PROPOSED STANDARD.

SHENKER, S. ET WROCLAWSKI, J. (1997a). *RFC2215 : General Characterization Parameters for Integrated Service Network Elements.* IETF. Status : STANDARDS TRACK.

SHENKER, S. ET WROCLAWSKI, J. (1997b). *RFC2216 : Network Element Service Specification Template.* IETF. Status : INFORMATIONAL.

SOLIMAN, H., CATELLUCCIA, C., EL MALKI, K. ET BELLIER, L. (2005). *RFC4140 : Hierarchical Mobile IPv6 mobility management (HMIPv6).* IETF. Status : EXPERIMENTAL.

TALUKDAR, A., BADRINATH, B. ET ACHARYA, A. (2001). MSRVP : a Resource Reservation Protocol for an Integrated Services Network with Mobile Hosts. *Wireless Networks*, vol. 7, pp. 5–19.

TANENBAUM, A. (2002). *Computer Networks (fourth edition).* Prentice-Hall International, Inc.

TSENG, C., LEE, G., LIU, R. ET WANG, T. (2003). HMRSVP : a Hierarchical Mobile RSVP Protocol. *Wireless Networks*, vol. 9, pp. 95–102.

WANG, Z. (2001). *Internet QoS - Architectures and mechanisms for Quality of Service.* Morgan Kaufmann.

WROCLAWSKI, J. (1997a). *RFC2210 : The Use of RSVP with IETF Integrated Services.* IETF. Status : PROPOSED STANDARD.

WROCLAWSKI, J. (1997b). *RFC2211 : Specification of the Controlled-Load Network Element Service.* IETF. Status : STANDARDS TRACK.

YADAV, S., YAVATKAR, R., PABBATI, R., FORD, P., MOORE, T., HERZOG, S. ET HESS, R. (2001). *RFC3182 : Identity Representation for RSVP.* IETF. Obsoletes RFC2752. Status : STANDARDS TRACK.

ZHANG, L., PIERRE, S. ET MARCHAND, L. (2005). Optimization of Handover Performance for FMIPv6. *Intelligence in Communication Systems.* pp. 169–178.